



UNIVERSIDAD  
DE LA REPUBLICA  
URUGUAY



UNIVERSIDAD DE LA REPÚBLICA  
FACULTAD DE INGENIERÍA

Tesis para optar por el título de  
Magíster en Computación

EVALUACIÓN DE CALIDAD DE VIDEO EN UNA  
APLICACIÓN P2P: GOALBIT

Autor: NICOLÁS DE LEÓN

Director de Tesis: PABLO RODRÍGUEZ-BOCCA

Montevideo, Uruguay

2010

## Contenido

I	Introducción .....	4
I.1	Objetivo .....	5
I.2	Motivación .....	6
I.3	Contribución de este trabajo .....	6
I.4	Estructura y resumen del trabajo .....	7
II	Conceptos de video y GoalBit .....	10
II.1	Video digital y estándares MPEG .....	10
II.2	La plataforma GoalBit .....	12
II.2.1	Estructura .....	12
II.2.2	Funcionamiento .....	13
II.3	Técnicas de mejora de calidad de video en redes P2P .....	16
III	Métricas y factores incidentes en la calidad de video .....	20
III.1	Valoración subjetiva de calidad .....	20
III.2	Valoración objetiva de calidad .....	23
III.3	Pseudo-Subjective Quality Assessment (PSQA) .....	24
III.3.1	Etapas generales de la metodología PSQA aplicada .....	25
III.4	Factores que inciden en la calidad .....	26
III.5	Caracterización de las redes de transporte .....	27
III.5.2	Acceso xDSL .....	27
III.5.3	Acceso celular .....	28
III.5.4	Herramienta de modelado de la red .....	30
IV	Aplicación de la metodología PSQA .....	32
IV.1	Configuración de la red de prueba con GoalBit .....	32
IV.2	Descripción general de la metodología .....	32
IV.3	Procedimiento paso a paso .....	35
IV.4	Observaciones sobre el proceso .....	43

---

V	Análisis de resultados .....	44
V.1	Formulando la calidad de experiencia .....	45
VI	Conclusiones .....	50
	Referencias .....	52
	Glosario .....	59
	Anexo 1 Configuración de nodo con NetEm2 .....	60
	Anexo 1 Configuración de nodo con NetEm2 .....	60
	Requisitos de NetEm2.....	60
	Ejecución de NetEm2.....	62
	Anexo 2 Datos de las muestras de video .....	63
	Parámetros de las muestras de video .....	63
	Evaluaciones.....	66

## Extracto

Este trabajo explora el terreno de la Calidad de Experiencia (QoE) en la distribución de contenido de *streaming* de video en formato H.264 sobre una red *peer-to-peer* (P2P) y en particular usando la aplicación GoalBit. Se aplica la metodología PSQA (*Pseudo Subjective Quality Assesment*) para evaluar la calidad frente a condiciones de red precarias, como un ancho de banda limitado, retardo y pérdidas, en redes de acceso y transporte típicas como ADSL (*Asymmetric Digital Subscriber Line*), 3G UMTS (*Universal Mobile Telecommunications System*) o HSPA (*High Speed Packet Access*).

Traduciendo los parámetros del espacio de variables de red mencionados al de las variables a nivel de aplicación (pérdidas por tipo de cuadro) se obtiene como resultado una metodología sencilla cuyos resultados pueden extenderse a diversos contextos de red y/o sistemas de distribución de flujos de video.

## Palabras clave

PSQA, P2P, calidad, video, 3G, ADSL, GoalBit.

## I Introducción

Internet puede considerarse como un gran repositorio de contenidos, utilizado por usuarios que pueden ser a la vez proveedores y consumidores de contenido diverso y de calidad variable. Este contexto fundamenta la necesidad de evaluar la calidad desde la experiencia o percepción del usuario, conocida como QoE. Esa medida de calidad puede ser clasificada como objetiva o subjetiva, pero en lo que parece haber un consenso general [76, 98, 101] es que es fuertemente dependiente del contenido del que se trate. En el campo de video se ha analizado la QoE para redes tan variadas como IPTV en [95], video en redes celulares [77, 99, 111], o sobre redes *Peer-to-Peer* (P2P) en [1], las cuales son el foco específico de este trabajo.

De las diversas opciones P2P, aquí se presentan resultados para GoalBit [4]. En el Capítulo II se presentarán las características principales de esta aplicación, la cual se basa en el protocolo BitTorrent [107]. Si bien BitTorrent no es el único protocolo P2P, es uno de los más utilizados. A modo de referencia de uso de P2P y en particular asociado al video, en [75] se evalúa el uso y la performance de BitTorrent encontrando que en el 2004 habían casi 600.000 descargas diarias, de las cuales casi la tercera parte es contenido de video. Este mismo estudio observa la degradación de calidad asociada a fallas frecuentes de red, como fallas del *Tracker* (servidor que mantiene referencias de contenido, ver II.2 para mayor detalle) o de servidores espejo o fallas del transporte.

En este trabajo se evalúa el impacto que tiene la red de transporte sobre la calidad de flujos de video codificados según estándar ITU H.264 o MPEG-4 parte 10 o MPEG4-

AVC [30], cuyas características principales se encuentran en el Capítulo II, sección II.1. El impacto en calidad es evaluado concretamente en términos de ancho de banda limitado, el retardo y su variación (jitter), las pérdidas y las ráfagas. Si bien existen antecedentes de estudios similares, la ventaja de esta aproximación es que se plantea una metodología sencillamente reproducible y extensible como es PSQA (*Pseudo Subjective Quality Assessment*) [24, 25].

PSQA ha sido aplicada en diversos estudios relacionados con el presente, como se explica en la sección I.3 y su instanciación para este trabajo se describe en el Capítulo III. En particular este trabajo es el primero en profundizar en la aplicación de PSQA a video con codificación H.264, la cual es la más difundida en sistemas profesionales.

Se opta por estudiar el impacto en la calidad de video distribuido en redes de transporte frecuentes en la actualidad, planteando una abstracción a nivel de los parámetros que afectan la calidad lo cual permitirá la extrapolación a otros contextos de red. Como sistema de distribución se utiliza una red P2P particular, que es GoalBit y como redes de transporte se recurre a una combinación de red cableada con inalámbrica, siendo ADSL (Asymmetric Digital Subscriber Line) [20, 21] el exponente cableado y 3G UMTS (*Universal Mobile Telecommunications System*) [22, 23] con HSDPA (High Speed Downlink Packet Access) [79, 80, 81] el ejemplo de red inalámbrica. La idea de esta mezcla es plantear un escenario híbrido frecuente en Uruguay y en muchos países, a tasas de transferencia consistentes para ambas y aceptables para la aplicación (se trabaja a 512 Kbps).

Este trabajo es extensible porque si bien se analiza el impacto de la variación de los parámetros de red como retardo, variación del mismo, pérdidas y ráfagas de pérdidas, se mapean a pérdidas a nivel de aplicación, como son los tipos de cuadros H.264. De esta manera se logra analizar la calidad esperada en función de las pérdidas observadas a nivel de cuadros H.264, siendo parámetros comprensibles (y eventualmente controlables) por el sistema de distribución de video.

## ***I.1 Objetivo***

El objetivo de este trabajo es evaluar la calidad de experiencia de *streaming* de video en una red P2P cuando es implementada sobre una red de acceso ADSL o UMTS/HSDPA de calidad variable en cuanto a retardo, variación de retardo, pérdidas y ráfagas de pérdidas. En particular está centrado en una aplicación concreta como es GoalBit, como extensión a dicho proyecto. Para dicho objetivo se recurre a herramientas conocidas y en evolución, como Netem [6] y Netem2 [13] para modelar la red de transporte como se explica en III.5.4, la metodología PSQA como se explica en III.3 y las Redes Neuronales, cuyos fundamentos no se detallan en este trabajo por ser ya conocidas y utilizadas en varias publicaciones, siendo [1] y [96] las más relacionadas a este estudio. El resultado es una fórmula paramétrica capaz de predecir la calidad de experiencia de video en GoalBit.

## 1.2 Motivación

La popularidad de las aplicaciones P2P es indiscutible e Internet es hoy en día, entre tantas cosas, sinónimo de comunidad de intercambio multimedia. La penetración de acceso a Internet en los hogares es cada vez mayor y los anchos de banda se han superado enormemente, de manera de que hoy 1 Mbps [43, 44] se puede considerar un ancho de banda normal en Sudamérica. Entonces es posible plantear servicios multimedia como video *streaming*. Al mismo tiempo, considerando que la oferta de redes de servidores centralizados de video en vivo no es muy abundante por los costos involucrados de despliegue y mantenimiento, es cada vez más viable un escenario en el que esa transferencia sea sobre una red P2P, porque es un tipo de servicio al que la gente está acostumbrada. Si adicionalmente a esta red se la dota de una capacidad de predicción de la calidad de experiencia y eventualmente de herramientas correctivas, se transforma en un diferencial importante respecto a sus pares competidores en tipo de servicio como son las cadenas de cable TV y en tipo de transporte, como son otras aplicaciones P2P, cuya filosofía es mejor esfuerzo.

Por todo esto, la concepción de una fórmula de evaluación dinámica de calidad de servicio que se acople al mecanismo e infraestructura P2P tiene mucho sentido actualmente.

## 1.3 Contribución de este trabajo

Si bien otros autores ya han evaluado los impactos de la red de transporte en la calidad de video, los enfoques son ligeramente diferentes.

En [111] se realiza un estudio de la calidad de video en redes 3G-WLAN (Wireless Local Area Network), con foco principal en la transición entre redes. Se presenta un modelo probabilístico para la coexistencia de ambas redes y se evalúa la calidad percibida desde la óptica de relación señal-ruido de pico (PSNR). No obstante PSNR puede no ser fiel a la calidad percibida, podría entenderse como una medida aproximada. Precisamente en [77] se promueve la metodología PSQA como más específica para la evaluación de la calidad percibida, comparando sus resultados con la técnica PSNR en escenarios de red inalámbrica donde se varían las condiciones de red (como retardos, jitter y pérdidas).

La referencia más cercana al presente trabajo está dada por [1], ya que allí se aplica la metodología PSQA a una red P2P predecesora de GoalBit evaluando diferentes parámetros de calidad y se propone las pérdidas por tipo de cuadro como evaluadores candidatos de calidad. No se profundiza particularmente en video H.264 ni se extiende para redes inalámbricas celulares. Un complemento a este trabajo puede estar dado por [99], en donde se analiza la distribución unicast de video H.264 sobre redes UMTS con HSDPA, brindándose incluso recomendaciones a nivel de optimización de la red UMTS con HSDPA para mejorar la calidad percibida, pero no se evalúa el sistema de distribución de video (como P2P). Particularmente sobre la distribución de video en redes UMTS, [84] plantea el uso de los servicios de multicast y broadcast especialmente definidos para dicha red, los cuales permiten incrementar la capacidad

de transmisión de video. Un aporte interesante de este estudio es validar una tasa de 256 Kbps para la transmisión de video en usuarios de la red UMTS con HSDPA.

Este trabajo se considera por ende un aporte metodológico y a la vez una profundización en el relevamiento de la calidad distribución de video H.264 sobre redes ampliamente difundidas (P2P), en un transporte frecuente y compatible en tasa de transferencia como es ADSL y 3G con HSDPA para una tasa moderada como 512 Kbps.

Concretamente los aportes son:

- una profundización de la metodología PSQA con aplicación en un sistema de distribución de video P2P concreto como GoalBit, evaluando los factores de calidad más relevantes en dicha aplicación
- una aproximación a la QoE de video en redes ADSL y UMTS, modelada a través de NetEm2 [13], estudiando la percepción de calidad en un escenario híbrido de transporte de video
- la construcción de una red de prueba simple, pero flexible para la evaluación de calidad de video ante una red de transporte inestable
- la evaluación de flujos de video H.264, observando los cuadros más sensibles y los contenedores más apropiados. Los conceptos de cuadro de video y contenedor se explicarán en el Capítulo II.

#### ***1.4 Estructura y resumen del trabajo***

El trabajo se divide en seis Capítulos. Un primer Capítulo introductorio. El Capítulo II describe conceptos fundamentales de video digital (sobretudo MPEG-4 que es el que se utilizará en este trabajo) y un resumen de GoalBit como aplicación P2P, incluyendo al final una breve digresión para presentar algunas tendencias para la mejora de calidad en redes P2P. Luego el Capítulo III presenta algunos conceptos básicos y necesarios sobre calidad, explica la metodología seleccionada para la determinación de la calidad y analiza potenciales parámetros de calidad. El Capítulo IV describe el escenario de prueba construido y el procedimiento seguido en este trabajo. Finalmente en los Capítulos V y VI se presenta la evaluación de los resultados y las conclusiones.

El trabajo consiste en tres instancias:

- 1) Interiorización en video streaming, la plataforma GoalBit, la metodología PSQA y determinación de potenciales parámetros de calidad que afecten al flujo de video.
- 2) Generar una maqueta de pruebas para emular la red P2P: para ello se contará con un cliente, un servidor y una emulación de red ADSL o 3G. Para la emulación de red se utiliza Netem (Network Emulator) [6] y Netem2. Sobre éstas se analizará el impacto de la red de transporte en los parámetros de calidad definidos, a la vez que se validarán los mismos.

- 3) Aplicar la metodología PSQA: generando secuencias de prueba, presentándolas a un conjunto de observadores para obtener una medida tipo *Mean Opinion Score* (MOS) [36], filtrando las mismas y entrenando una Red Neuronal (RN) [26, 61, 62] con ellas para obtener una fórmula de calidad.

Como producto se tendrá una fórmula de evaluación dinámica de la calidad basada en un conjunto reducido de parámetros, la cual podrá ser integrada a GoalBit. Esta herramienta es la base de mecanismos de estadística y monitoreo de calidad de la red P2P, como se presentan en [66, 67]. Adicionalmente, en base a ella el servidor podría tomar acciones correctivas sobre la codificación, el fraccionamiento y/o la redundancia en esquemas multi-camino o multi-fuente, como se comentará en el Capítulo II, sección II.3.

---

- Página en blanco, intencional -

## II Conceptos de video y GoalBit

### II.1 Video digital y estándares MPEG

El video digital ha tenido desde sus primeras etapas en los 80's una gran evolución conceptual. Desde una primera aproximación de algo más que la digitalización de la señal a su preparación para la distribución, en donde consideraciones tales como ancho de banda y luego calidad cobraron un sentido especial. Es así que, buscando reducir el ancho de banda demandado para la transmisión, el concepto de compresión de video es inherente ya al video digital desde finales de los 80 y principios de los 90'. Quienes posibilitan esta compresión de video son los *codec* (denominados así por la cadena necesaria en lado trasmisor, *encoder* y en lado receptor, *decoder*) y entre ellos hay un grupo muy destacado que ha permanecido vigente desde sus inicios que son los del grupo MPEG (*Moving Picture Experts Group*).

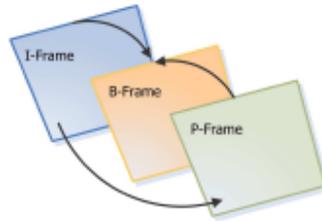
MPEG en realidad cubre una familia de estándares, aunque los que destacan actualmente son MPEG-2 y MPEG-4:

- MPEG-1 [27]: estándar inicial de compresión de audio y vídeo. Usado después como la norma para CD de vídeo, incluye popular formato de compresión de audio Capa 3 (MP3).
- MPEG-2 [28]: normas para audio y vídeo para difusión de calidad de televisión. Utilizado para servicios de TV por satélite como DirecTV (Cadena de televisión vía satélite) [114], señales de televisión digital por cable y (con ligeras modificaciones) para los discos de vídeo DVD (*Digital Versatil Disc*) [41, 42].
- MPEG-3 [29]: diseñado originalmente para HDTV (Televisión de Alta Definición), pero abandonado posteriormente en favor de MPEG-2.
- MPEG-4 [30]: expande MPEG-1 para soportar "objetos" audio/vídeo, contenido 3D, codificación de baja velocidad binaria y soporte para gestión de derechos digitales (protección de copyright).
- MPEG-7 [31]: sistema formal para la descripción de contenido multimedia
- MPEG-21 [32]: MPEG describe esta norma futura como un "cuadro multimedia".

En general el grupo MPEG ha trabajado en colaboración con ITU-T, siendo por ejemplo el estándar MPEG-4 parte 10 (ISO/IEC 14496-10) equivalente al estándar H.264, pero también son ejemplo de colaboración los estándares H.261 [33], H.262 [34] y H.263 [34].

MPEG-4 supera a MPEG-2 en eficiencia de compresión, por eso es el más adoptado actualmente, pero ambos se basan en un esquema de cuadros I-P-B, representando los "I" (Intra-frames) una codificación completa de una imagen, los "P" una predicción (Predicted-frames) compensando movimiento, basado en cuadros previos "P" o "I" y

los “B” (Bidireccional-frames) una interpolación basada en cuadros previos o futuros de “P” o “I”. La figura 2.1 muestra esta dependencia:



- Figura 2.1: Relación entre cuadros I, P y B en MPEG –

La compresión se obtiene de la redundancia codificada, de la generalmente alta correlación entre cuadros. Los cuadros I codifican redundancia espacial, mientras que los cuadros B y P codifican redundancia temporal. Debido a que los cuadros adyacentes son a menudo bien correlacionados, los cuadros P pueden ser del 30% del tamaño de un cuadro I, y el cuadro B al 15% de su tamaño [39]. Los cuadros I, P y B se agrupan en arreglos denominados GOP (Group Of Pictures), en general encabezados por un cuadro I. Arreglos típicos de MPEG-2 un GOP de 15 cuadros (como ser I\_BB\_P\_BB\_P\_BB\_P\_BB\_P\_BB [40], pero pueden ser de 12 y hasta de 20 cuadros) y para MPEG-4 de alrededor de 250 cuadros.

MPEG-4, inicialmente pensado para transmisión de video de baja calidad (8 bits/muestra) para videoconferencias y aplicación en Internet, planteó algunos cambios conceptuales, sin preocuparse por la compatibilidad hacia atrás:

- Fundamentalmente concibe objetos, conocidos como VOP (Video Object Plane) en lugar de imágenes como MPEG-2.
- Introduce dos nuevos tipos de cuadros, el SP (Switching P) y SI (Switching I), necesarios para codificar la transición entre flujos de videos usando predicciones en lugar de cuadros I, que son más pesadas.
- Agrega un filtro de desbloqueo, mejorando la percepción de calidad al reducir el efecto de “bloques” en el video.
- La compensación de movimiento se realiza analizando bloques y sub-bloques del cuadro (16x8, 8x16 o 8x8 píxeles y menores), llegando hasta precisiones de un cuarto de pixel.

Luego, al ver la escasa penetración del códec en mercados profesionales, MPEG-4 agregó entre sus perfiles de calidad algunos elevando la cantidad de bits por muestra y mejorando el esquema de muestreo, definiendo por ejemplo la extensión *High 4:4:4*, que soporta un esquema de muestreo 4:4:4, 12 bits/muestra (lo normal es 8 bits/muestra) y la codificación de regiones sin pérdidas. El esquema de muestreo refiere al submuestreo de la información de crominancia (Cb: azul y Cr: rojo) respecto de la de luminancia. Esto es posible debido a que el ojo humano es menos sensible a

la falta de información de color que a la falta de información de brillo o luminosidad. Entonces se puede sub-muestrear las componentes de color (en realidad en la notación se considera la suma de Cb y Cr) reduciendo el ancho de banda necesario para la transmisión del video. En [112] se hace una buena explicación de la notación y los conceptos correspondientes. A modo de referencia, con el perfil 4:4:4 las tasas de transmisión pueden variar en niveles de calidad desde 256 Kbps hasta 960 Mbps.

## II.2 La plataforma GoalBit

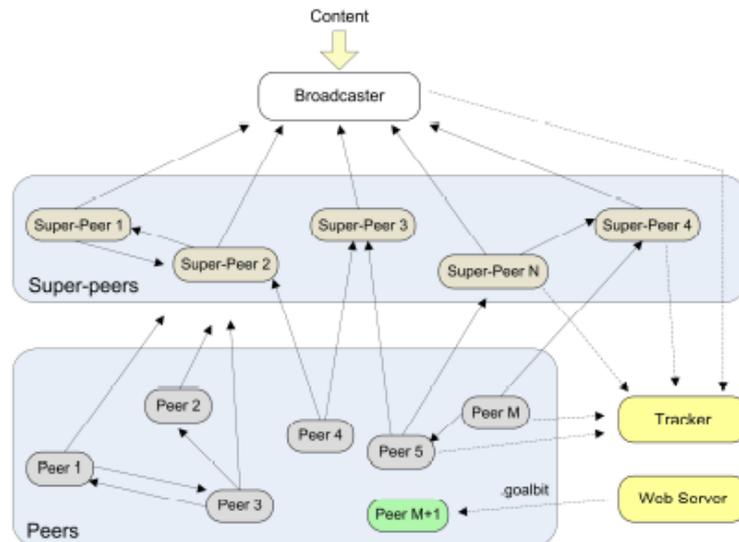
### II.2.1 Estructura

GoalBit es una aplicación de distribución de contenido multimedia *Peer-to-Peer* basada en el protocolo bitTorrent. Está fuertemente orientada a la distribución de video de alta calidad, tanto descarga como carga por aquellos que quieran contribuir a la comunidad GoalBit y apuesta a aplicar la metodología PSQA para cumplir con ese objetivo.

La arquitectura de GoalBit se presenta en la figura 2.2 y se basa en el modelo de bitTorrent en cuanto a compartir recursos se refiere, con algunos ajustes en el esquema de interacción entre nodos y jerarquizado en 3 tipos de nodos:

- ✓ **Broadcaster:** es quien cumple el rol de servidor central de distribución de video, compuesto por el *broadcaster server* y los *broadcaster-peers*. El primero obtiene, almacena y fracciona las secuencias de video en flujos o canales para entregárselos a los segundos, que son quienes distribuyen las piezas en la red.
- ✓ **Super peer:** Los *super-peers* son auxiliares del *broadcaster*, son clientes de gran porte y los únicos autorizados a comunicarse con los *broadcaster-peers*. Además pueden comunicarse entre sí y con los *normal-peers*.
- ✓ **Normal-peers o peers:** son los clientes de la red, son reproductores de video que tienen la capacidad e intercambiar piezas con otros *normal-peers* o con *super-peers*.

Además de los nodos mencionados, otra entidad presente en GoalBit es el *Tracker*, el cual se encarga de mantener la información de qué *peers* están viendo cada canal (y por ende son potenciales “servidores” o “fuentes” para esa secuencia). Esa información es utilizada para entablar la comunicación entre los nodos.



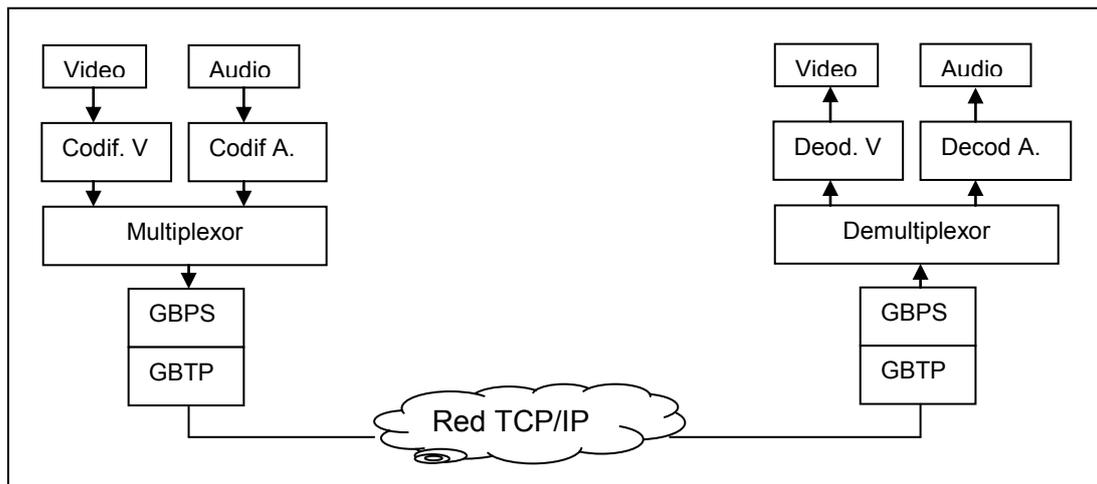
- Figura 2.2: Arquitectura de red GoalBit -

Los *Trackers* guardan la configuración de canales y las fuentes disponibles en archivos *.goalbit* (equivalentes a los *.torrent*). Para fomentar el uso de la red estos archivos son publicados y referenciados en servidores web haciéndolos accesibles a gran cantidad de usuarios.

Cabe observar que si bien la arquitectura mencionada es genérica y completa, GoalBit puede omitir o combinar algunos roles. Por ejemplo, GoalBit puede funcionar sin un *Tracker*, en lo que se denomina modo *Tracker-less*, usando una red basada en Kademila [106]. Si bien este puede tener algunos problemas de estabilidad y vulnerabilidades como plantea [58], es una opción flexible que con ciertas precauciones y/o configuraciones específicas [59, 60] puede alcanzar una buena eficiencia y estabilidad. Es a su vez posible prescindir del nivel de *super-peer*, fusionando este nivel o combinando sus funciones con el *broadcaster*.

## II.2.2 Funcionamiento

Cuando se habla de un flujo de video entre un servidor y un cliente en una red P2P está implícita una adaptación de la fuente y el fraccionamiento del mismo en lo que se denominan piezas de video o *chunks*. Cada fuente de video es descompuesta en la señal de video y de audio, los cuales una vez codificados deben ser encapsulados o multiplexados en un contenedor, que es lo que en realidad viaja por la red y se reproduce. El control del encapsulamiento y el fraccionamiento e identificación de piezas son aspectos esenciales cuando se trabaja con flujos de video en vivo. GoalBit introduce el GoalBit Packetized Stream (GBPS) para lo primero y el Goalbit Transport Protocol (GBTP) para lo segundo. La figura 2.3 muestra un esquema de la transmisión de flujo de video y seguidamente se describen GBTP y GBPS con mayor detalle.

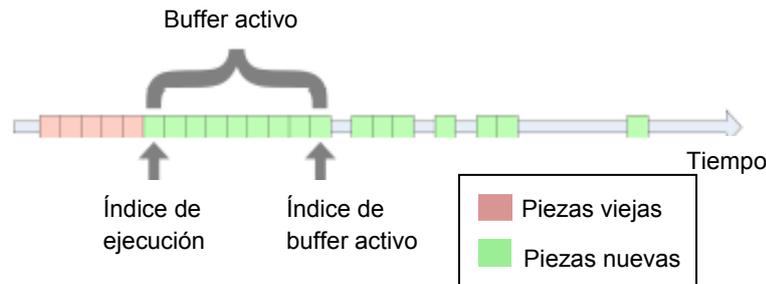


- Figura 2.3: Trasmisión de flujo de video en una red P2P GoalBit -

El GBPS se introduce casi al final de la cadena de distribución de video, tomando la salida de los multiplexores de video antes de entregarla al protocolo de transporte correspondiente. Algunos multiplexores usuales para flujos de video son *MPEG Transport Stream (MPEG-TS)*, *MPEG Packet Stream (MPEG-PS)* o *Advanced System Format (ASF)*. Los protocolos de transporte pueden ser RTP (*Real Time Protocol*), HTTP, UDP o en el caso de GoalBit el GBTP. La elección del contenedor y el protocolo de transporte es generalmente configurable en la aplicación P2P, pero MPEG-TS o ASF y GBTP/TCP son lo más usado en GoalBit. En [54], un paper avalado por Envivio [55] presenta una comparación de la trasmisión de video MPEG-4 sobre RTP nativo y sobre MPEG-TS observando ciertas ventajas de RTP, pero igualmente depende generalmente del contexto de aplicación la mejor opción.

Un diferencial importante de GBPS es que es un soporte multi-protocolo de multiplexado, puesto que incorpora identificación (presenta un encabezado de multiplexor), marcado (inicio y fin de contenido video multiplexado) y fragmentación, para ajustar las tasas variables de los diferentes multiplexores a paquetes de longitud fija

En lo que respecta al GBTP, como muestra la figura 2.3, se ubica sobre el grupo de protocolos TCP/IP [45, 46], siendo como un adaptador de TCP para trasmisión de flujo de video. Incorpora características específicas para flujos de video, tales como una identificación de la pieza de video o *chunk*, una ventana deslizante en el cliente y un *buffer activo* (ver figura 2.4), también en el cliente. Las piezas a transmitirse son identificadas con un número secuencial (en forma cíclica) y el cliente consume esas piezas en forma consecutiva dentro de un rango o ventana deslizante (a medida que se reconocen las piezas recibidas). Ese conjunto de piezas consecutivas recibidas conforma lo que se conoce como *buffer activo*, cuyo extremo superior esta dado por el *índice de buffer activo (ABI)*, que es el identificador de pieza más alto recibido.



- Figura 2.4: Modelo de buffer activo de cliente GoalBit -

El *Tracker* guarda la información de cada canal y de todos los clientes conectados. Cada canal tiene un hash asociado junto con datos estáticos y dinámicos de los clientes. Dentro de los datos estáticos se encuentra el tipo de *peer*, la IP y el puerto utilizado para la conexión. Los datos dinámicos contienen el ABI y eventualmente información de calidad de experiencia (QoE), medida por el cliente.

La información de canal está contenida en los archivos *.goalbit*, cuyo contenido se muestra en la tabla 2.1.

Parámetro	Descripción
<b>channel id</b>	Identificador de canal (string)
<b>chunk size</b>	Tamaño de pieza de video
<b>Tracker url</b>	URL del <i>Tracker</i> donde se ejecuta el canal
<b>bitrate</b>	Tasa de transferencia de streaming
<b>name</b>	Nombre del canal
<b>description</b>	Descripción del canal
<b>Thumb</b>	Logo del canal

- Tabla 2.1: parámetros de canal incluidos en el archivo goalbit -

Cada *peer* debe reportarse al *Tracker* al inicio y luego periódicamente mientras permanezca activo en la red GoalBit en mensajes denominados *anuncios*. El propósito del mensaje de anuncio inicial hacia el *peer* es obtener la lista de peers fuente del video buscado. El *peer* solicitante indica un índice (mínimo) de piezas requeridas (numwant). En los anuncios periódicos (cuya frecuencia controlada por el *Tracker*) el objetivo es que el *Tracker* mantenga información de estado de cada *peer*, por lo que el *peer* indica el ABI y la QoE, pudiendo agregar la solicitud de *peers* fuente adicionales para el flujo de video que este viendo.

Una vez que el *peer* ha establecido la comunicación inicial con el *Tracker* y tiene el dato de *peers* con los que se debe comunicar para obtener el flujo de video buscado, la comunicación prosigue con dichos *peers* usando TCP. También en esta comunicación se distinguen dos tipos de mensajes:

- Mensajes de contexto: Informan estado (ej. conectado) y capacidades del *peer* como proveedor de piezas de video. Existen 5 mensajes de este tipo: HAND-SHAKE, BITFIELD, HAVE, WINDOWUPDATE and KEEP-ALIVE.
- Mensajes de intercambio: son los usados para intercambiar piezas. Existen 8 mensajes de este tipo: INTERESTED, NOT INTERESTED, CHOKE, UNCHOKE, REQUEST, CANCEL, PIECE and DONT HAVE.

De los mensajes de contexto, quizás el más interesante para un cliente goalbit es el Bitfield, el cual informa las piezas con las que un *peer* cuenta. Los mensajes de intercambio por su parte son bastante sugestivos por su nombre y en este caso todos brindan información igualmente interesante. Para obtener una información más completa del contenido de los mensajes entre *Tracker* y *peers* o entre *peers* consultar [4].

Dos principios mantienen el correcto equilibrio en la red GoalBit: el criterio de selección de *peers* y el criterio de selección de piezas. El criterio de selección de *peers* (evaluado en [64]) sigue un principio de ley del tali3n como BitTorrent, es decir que se habilita para descargar a aqu3l *peer* que m3s comparte, pero la diferencia es que al inicio se selecciona aleatoriamente un *peer* con el cual compartir sin esperar retribuci3n de generosidad. Esta estrategia se denomina *optimistic-unchocking* o desbloqueo optimista. El criterio de selecci3n de pieza por su parte fue estudiado para GoalBit en [63]. Se trata de un compromiso entre el criterio tradicional de los sistemas *peer-to-peer* (la m3s rara primero) y el m3s eficiente para flujos de video, que es la avaricia de piezas (elegir la pr3xima que el cliente necesita). Un criterio tradicional de *peer-to-peer* aplicado a flujos de video causa una gran demanda de buffer inicio, por lo que no es lo mejor. Mientras que un criterio avaricia permite optimizar la vista de un cliente individual, quiz3s a costa de que no todos puedan completar la descarga. Por eso un equilibrio entre ambos criterios es lo que logra una distribuci3n de piezas m3s uniforme en la red, que es lo que se busca como meta. GoalBit administra este criterio con tres tipos de buffer: urgente, normal y futuro, que se van completando en orden, siguiendo una distribuci3n exponencial decreciente (ver detalles en [4]).

### II.3 T3cnicas de mejora de calidad de video en redes P2P

Hay dos grandes tendencias en lo que respecta a la optimizaci3n de la trasmisi3n de flujo de video: multi-fuente o multi-camino. Estas dos opciones han sido evaluadas en diferentes campos de telecomunicaciones y ambas son v3lidas dependiendo de la capa (OSI) en la que trabajemos y el grado de control que se disponga en dicha capa.

Una opción multi-camino o diversidad multi-camino es muy empleada en capas bajas, desde la capa física hasta IP. Ejemplos de control en capa física puede ser en redes celulares o WLAN la diversidad de antenas y las técnicas Multiple-Input Multiple-Output (MIMO) [50, 51, 82]. Ejemplos de control en IP son esquemas anycast o enrutamientos redundantes y basados en métricas de calidad específicas. Pero no siempre es posible (o es muy costoso) controlar las capas bajas de la red, a veces sólo se es usuario de esa red, que se usa como transporte. Técnicas multi-camino más complejas involucran codificación adaptativa y el uso de Multi Description Coding (MDC). En [52, 53] se evalúan mecanismos de ecualización de la calidad en una red restrictiva y en [97] se presenta una ecualización basada en medidas de QoE y estimación de ancho de banda disponible.

La tendencia de multi-fuentes puede ser revalidada para redes P2P como una “multi-recepción”, ya que en realidad lo que interesa es la percepción del lado del cliente o en términos de la red P2P, del *peer*. De hecho en [74] se plantea un mecanismo multi-fuente, que si bien implementa un protocolo y arquitectura específica, está enfocado desde el lado receptor.

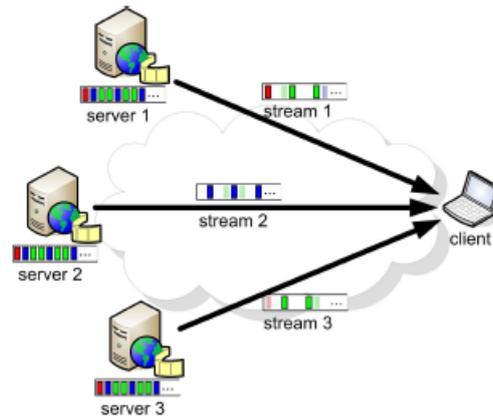
Entendiendo multi-fuente como la capacidad del sistema de brindar más de una fuente simultánea para un cliente dado y la posibilidad de éste de capturar el flujo de más de una fuente, es posible validar esta idea para redes P2P GoalBit pensando la cadena de *broadcaster*, *super-peer* y *normal-peer(s)* hasta el *peer* cliente final como un “servidor” o “fuente” equivalente, con una caracterización de pérdidas, retardos y demás. Es por eso que se comenta esta tendencia con un poco más de detalle.

Un esquema multi-fuente (en la nueva concepción planteada, como multi-recepción) busca incrementar la probabilidad de un cliente de completar su descarga. Si bien esto es asumido en una red P2P, es más complejo cuando se trata de flujos de video en vivo, puesto que el orden de los chunks es fundamental y la pérdida de algunos redundancia en mala calidad o directamente imposibilidad de reproducir el video (que no pasa con otros datos sobre redes P2P). Las aproximaciones al problema de multi-fuentes son variadas, pero sólo se comentarán comparativamente tres alternativas evaluadas en [1, 65], que son las que podría implementar GoalBit en un futuro próximo:

**Fraccionado:** Obviando el esquema de una única fuente, el esquema de fraccionado de fuentes es una alternativa sugestiva, pensando en las ventajas del multi-camino. El flujo de video se fracciona y se reparte entre diversos servidores/clientes, sin redundancia.

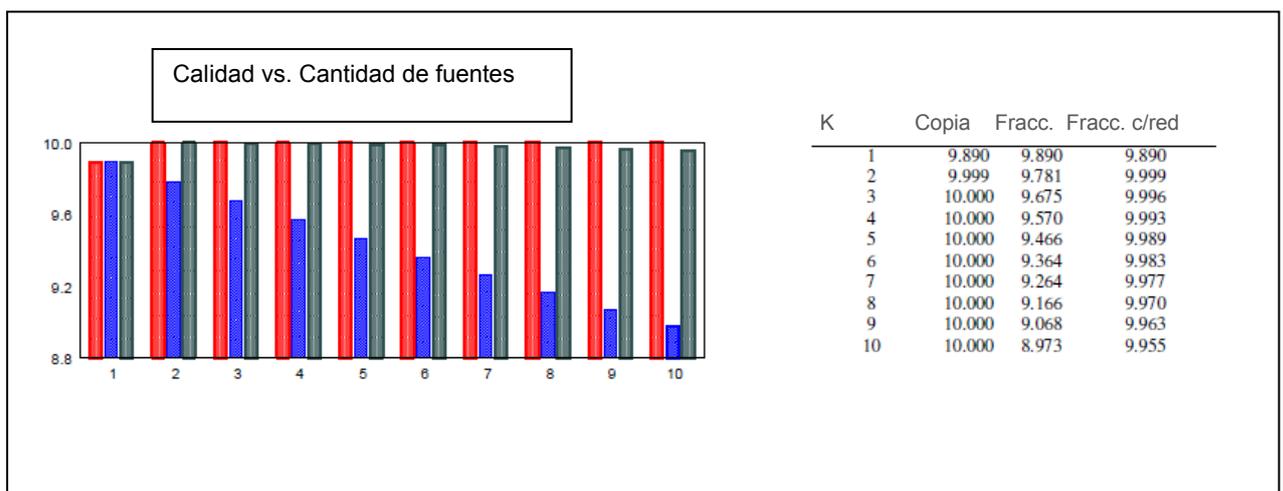
**Multi-fuentes con copia:** En este esquema se replica el flujo completo de video en varios servidores/clientes. Tiene la desventaja del ancho de banda, dado que es múltiplo de la cantidad de servidores en los que se copia.

**Fraccionado con redundancia:** Es un esquema intermedio, en el cual el flujo de video se fracciona de forma tal que siempre existen al menos dos servidores con un fragmento dado (ver figura 2.5).



- Figura 2.5: Multi-fuentes, esquema de fraccionamiento con redundancia -

En [1] se comparan estáticamente los enfoques variando la tasa de pérdidas (LR) y el tamaño promedio de las ráfagas de pérdidas (MLBS), observando que el mejor método es el de la copia, si bien el método de fraccionamiento tiene la ventaja de ser agnóstico al MLBS. La copia provee redundancia en todo momento, de ahí que la calidad percibida es muy buena, mientras que tanto una única fuente como el fraccionamiento en múltiples fuentes sin redundancia son sensibles a pérdidas o fallas en algún servidor del camino. Entonces se puede inferir que un método intermedio como el de fraccionamiento con copia parece el más indicado. Pero una evaluación estática cuando se trata de una red P2P no es suficiente, es necesario considerar el dinamismo de los *peers* de la red, modelar la desconexión de servidores de la red P2P. Una vez que el modelo está completo la pregunta evidente es ¿hasta cuándo es conveniente agregar fuentes? Cuantas más fuentes mayor la probabilidad de un cliente de obtener el flujo de video y en forma continua, pero el dinamismo de conexiones y desconexiones en la red P2P puede dificultar mucho la disponibilidad de fragmentos de flujo de video redundantes en todo momento, por lo tanto hay un equilibrio en función de la calidad percibida deseada. Nuevamente extrayendo las conclusiones de [1], la figura 2.6 resume el impacto en la calidad percibida para cada modelo conforme aumenta la cantidad de fuentes simultáneas.



- Figura 2.6: Calidad vs. Nro. de fuentes simultáneas en modelos multi-fuente según [1] -

Para GoalBit se considera como modelo de base el de fraccionamiento con redundancia basado en estas conclusiones, no obstante se observan los siguientes aspectos a tener en cuenta:

- Heterogeneidad de *peers*: un fraccionamiento equitativo sin contemplar las capacidades individuales de cada *peer* puede no ser óptimo, por lo tanto es recomendable incorporar un peso ponderado de reparto por “clases” de *peers* (notar que la determinación de los “mejores” *peers* puede no ser trivial, pero puede sustentarse inicialmente en estadísticas)
- Discriminación por tipo de cuadros: Basándose en los formatos más frecuentes de video, como son MPEG-2 y 4, es posible plantear una discriminación en función de los 3 tipos de cuadros representativos (I, P y B). Un esquema de fraccionamiento y redundancia ponderado por tipo de cuadro, en función del impacto de pérdida de cada tipo de trama resulta un camino razonable. Por ejemplo es razonable pensar en una redundancia mayor de los cuadros tipo I y P que los B, basados en la proporción media que estos tiene en la secuencia de video.

En [1] se plantea un modelo concreto para determinar una configuración multi-fuente óptima considerando los dos aspectos antedichos que puede ser trasladado a la aplicación GoalBit.

### III Métricas y factores incidentes en la calidad de video

Existen dos grandes tendencias para la evaluación de la calidad de video: la evaluación subjetiva (ver III.1) y la objetiva (ver III.2). La evaluación subjetiva consiste en la evaluación realizada por individuos, generando una puntuación determinada (en general debe existir una referencia de calidad). La evaluación objetiva intenta abstraerse completamente de la participación de los individuos, construyendo modelos teóricos y algoritmos de evaluación de calidad mayormente en base la distorsión percibida de la señal original.

Las metodologías subjetivas clásicas contempladas en general por la ITU [2] están orientadas a comparación de calidad con un cierto modelo de servicio inherente, que en general es la televisión. Además plantean ambientes de prueba de laboratorio, pensados ya sea para un ajuste inicial de la red de transmisión de video o ante eventuales fallas, pero no contemplan el dinamismo de la calidad de video. Un concepto más actual y específico que la calidad de servicio es la calidad de experiencia, que es lo que analiza este trabajo. Tiene sentido hoy en día analizar la evolución de la calidad de flujo de video desde la óptica del usuario y del contenido, como se expresa en [76, 98, 101]. Esto permite una realimentación directa entre el proveedor de contenido y el usuario, que lleva a la mejora de la red, a la vez que también permite tener una realimentación de la situación de la red para el proveedor de red o diseñador del transporte. Usuarios satisfechos en un contexto de red P2P redundan en una red más extensa y rica en contenido, lo cual es ciertamente una realimentación positiva para el proveedor de servicio, dicho esto en el más amplio de los sentidos de "servicio".

PSQA ha sido validada como metodología para determinar la calidad de experiencia [1, 77], incluso en redes celulares. Para recurrir a ella es necesario evaluar potenciales parámetros, como se explicará en las secciones III.4 y III.5, los cuales se irán refinando en función del contexto de aplicación como se plantea en V. No obstante, como la determinación de parámetros de entrada de la metodología PSQA es un procedimiento iterativo hasta converger en los parámetros que se consideran más relevantes, el Capítulo V retomará las ideas del presente Capítulo a la luz de los resultados de pruebas obtenidos.

#### III.1 Valoración subjetiva de calidad

La valoración subjetiva es sin duda la más natural de las evaluaciones, porque en definitiva la calidad de video es subjetiva, lo que interesa es la percepción del usuario sobre la imagen recibida. Pero a la hora de evaluar un sistema, es necesario darle a esa valoración un entorno formal, un procedimiento. La ITU formaliza algunos métodos de evaluación subjetiva en su estándar ITU-R BT.500-11 [2], de los cuales comentaremos tres que pueden ser aplicados a *streaming* de video.

- **Single Stimulus (SS)**
- **Double Stimulus Impairment Scale (DSIS)**
- **Double Stimulus Continuous Quality Scale (DSCQS)**

Todos los métodos tienen como resultado una valoración en escalas graduadas, que lo que representan es una analogía con el tradicional MOS (Mean Opinion Score) [36] de sistemas de telecomunicaciones, aunque pueden presentar 5 (como aquél) o más niveles.

El método que se describe con mayor detalle por ser el seleccionado para este trabajo es el de estímulo único (SS), o también conocido como método de puntuación por categorías absolutas (Absolute Category Ranking). Se ha elegido este método por su simplicidad y a la vez porque es el que más se acerca a la condición usual de evaluación de un observador de video en vivo: evaluación de calidad de una vez, dinámica.

El mismo consiste en la evaluación de un conjunto de secuencias de video con las siguientes características:

- ✓ Duración de cada secuencia: de aproximadamente 10s
- ✓ Intervalo entre secuencias: escena gris, de no más de 500 ms
- ✓ Cantidad de secuencias: No hay un valor explícito, sugerido entre 60 y 100 secuencias.
- ✓ Modalidad:
  - Se organizan 3 sesiones de no más de 30 min cada una, donde cada sesión es iniciada con un título.
  - La primera sesión debe incluir algunas escenas de acostumbramiento, unas cinco son suficientes (sus evaluaciones se descartan).
  - Se pueden repetir secuencias en diferentes sesiones para corroborar la coherencia del observador.
- ✓ Cantidad de observadores: entre 4 y 40, típico 15.
- ✓ Puntuación de la calidad: en escala de 5, 9 u 11 puntos. En definitiva la precisión de las 3 es la misma, son 5 valoraciones, por eso en este trabajo consideramos la de 5 puntos (ver figura 3.1).

Puntaje	Apreciación o juicio
5	Excelente
4	Bueno
3	Aceptable
2	Pobre
1	Malo

- Figura 3.1: Escala de puntuación en evaluación SS -

El método DSIS es utilizado para evaluar el impacto de un nuevo sistema a prueba o la red de transmisión utilizada (la salida de ello es el “video deteriorado” a evaluar). El coordinador de las pruebas deberá definir adecuadamente las condiciones o el procedimiento de prueba, el material (videos a evaluar) y el conjunto de parámetros a ser afectados acorde a los aspectos que se pretenden que se desea evaluar.

El video que ven los observadores es una secuencia intercalando el video original con el video deteriorado, pudiendo agregarse una imagen gris entre ambos. Los observadores deben evaluar el grado de deterioro observado en la señal en una escala de cinco niveles como indica la figura 3.2.

Puntaje	Apreciación o juicio
5	Imperceptible
4	Perceptible, pero tolerable
3	Un poco molesto
2	Molesto
1	Muy molesto

- Figura 3.2: Escala evaluación DSIS –

El método DSCQS es similar al anterior, pero se utiliza cuando no es posible disponer de un conjunto completo de material y/o parámetros de prueba y por ende sólo permite analizar un rango acotado de variación de calidad. En ese caso se utiliza un sistema que permite al/los observador/es alternar entre el video original y el distorsionado (salida A o B) hasta que considere que tiene la valoración (dentro de un plazo pre-establecido en el procedimiento). La evaluación es ingresada en una escala doble como la de la figura 3.3.

Valoración	Salida A	Salida B
Excelente		
Bueno		
Aceptable		
Pobre		
Malo		

- Figura 3.3: Escala evaluación doble DSCQS -

### III.2 Valoración objetiva de calidad

A veces es deseable que la valoración de calidad sea independiente de los individuos y a la vez valorando las ventajas de procedimientos automatizados, la comunidad científica ha investigado en lo que se denomina valoración objetiva de la calidad. Se trata de mecanismos que como base se independizan del observador, pero no así de una fuente de referencia y una cierta moderada o alta de cálculo.

En este trabajo se mencionan 2 de referencia, el primero por su simplicidad y el segundo por su mejor ajuste con la valoración de los usuarios.

- **Peek Signal to Noise Ratio (PSNR)**
- **Color Moving Picture Quality Metric (CMPQM)**

El método PSNR recoge métricas clásicas de telecomunicaciones, como son el cociente de potencias de señal original y error y el cálculo de mínimos cuadrados. Con una señal dada, lo que se busca es minimizar la diferencia de las funciones de luminancia en cada cuadro entre la secuencia original y la del video distorsionado. Si bien este mecanismo es simple, no es muy riguroso para el caso de video con color, dado que no considera color ni aspectos de la percepción humana. El alcance de PSNR es evaluado en [49].

El método CMPQM [3] por su parte incorpora las características fundamentales del modelo de Sistema de Visión Humana (HVS) [47, 48]. Es el sucesor, la extensión natural considerando el color del método "Moving Picture Quality Metric (MPQM)", ambos desarrollados en la institución francesa "École Polytechnique Fédérale de Lausanne" (EPFL). El modelo HVS considera alinealidades en el sistema visual de los seres humanos, tanto en el dominio de frecuencia espacial como temporal. Es así que por ejemplo los mismos estímulos se perciben diferentes cuando ocurren en espacios planos de cuando ocurren en bordes.

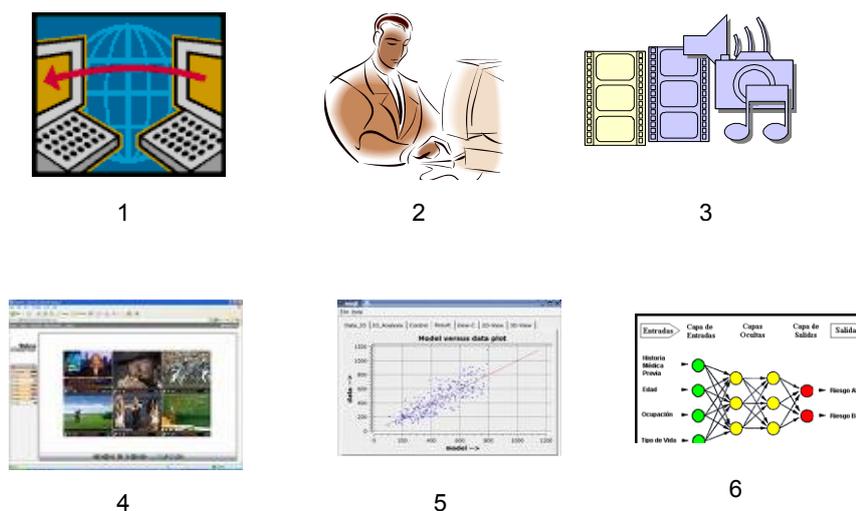
MPQM introduce la mayor parte del fundamento, como la importancia de un umbral mínimo de contraste para la detección de un cambio en la imagen (por parte del ojo humano). En su expresión matemática modela la calidad de video como una sumatoria ponderada de términos que valoran la percepción humana y modela la aproximación entre las señales original y distorsionada como una variante de PSNR considerando la distorsión global.

Este modelo presenta una buena correlación con las valoraciones subjetivas fundamentalmente en codificaciones de alta velocidad.

### III.3 Pseudo-Subjective Quality Assessment (PSQA)

La metodología PSQA combina lo mejor de ambos mundos: la valoración subjetiva y un mecanismo semi-automático de cálculo y ajuste de calidad. Para ello este trabajo aplica PSQA con las siguientes etapas generales, siguiendo el modelo de [1, 24, 86]:

- 1) Definición y construcción de un entorno de pruebas
- 2) Selección de parámetros representativos de la calidad y pre-validación de los mismos en plataforma de prueba
- 3) Generación de secuencias resultantes de la variación (controlada) de los parámetros elegidos en 2: muestras de video.
- 4) Evaluación de las secuencias obtenidas en 3 por un conjunto de observadores
- 5) Procesamiento de resultados de evaluación subjetiva
- 6) Entrenamiento de una Red Neuronal (en general aleatoria, RNN) con los resultados observados



- Figura 3.4: Etapas en la metodología PSQA: (1) Construcción del entorno de pruebas, (2) Selección de parámetros, (3) Selección de muestras de video afectado, (4) Observación y ranking, (5) Procesamiento de evaluaciones, (6) Entrenamiento de una Red Neuronal -

En pocas palabras PSQA conjuga la experimentación (para obtener una fórmula inferida de calidad) con la evaluación subjetiva (como validación de la fórmula) con técnicas conocidas y eficientes computacionalmente de entrenamiento como RNN. En este sentido es un mecanismo pseudo-subjetivo, porque las redes neuronales emulan la evaluación humana una vez entrenadas y semi-automático, porque persiste la instancia de validación humana, ya sea en el conjunto de observadores o el propio coordinador del proceso a la hora de filtrar y entrenar a la RNN o después para ajustar la fórmula si fuera necesario.

### III.3.1 Etapas generales de la metodología PSQA aplicada

Si bien el Capítulo IV describe en detalle el procedimiento aplicado, en esta sección se explican las seis etapas planteadas antes a modo de una primera aproximación a la metodología aplicada.

La definición y construcción del entorno de pruebas consiste en la instalación de un servidor de flujo de video P2P, un *Tracker* y un cliente interconectados en una red de pruebas. La red debe modelar la situación observada en redes reales, por lo que se recurre a aplicaciones como Netem, NIST Net [69], DummyNet [70], NS [71] u otros, buscando reproducir las tasas de pérdidas o retardos encontrados en la realidad.

En este trabajo se considera la red GoalBit, por lo que la maqueta de pruebas incluye un servidor que contiene el *Broadcaster* y el *Tracker* y otra máquina con un *peer* GoalBit. La red se emula en un router con Netem y Netem2, seleccionado por su integración nativa en Linux y su amplio uso.

La selección de parámetros que afectan a la calidad es un proceso iterativo, en el cual se plantearán potenciales parámetros que serán evaluados primeramente por quien lleva adelante la investigación y si resultan a priori relevantes, serán agregados a la lista de parámetros a evaluar por los observadores. En este trabajo, partiendo de las conclusiones obtenidas en [1] se considera la tasa de pérdida de cuadros I, P y B, la pérdida de red y la correlación entre pérdidas de red o ráfagas de pérdidas.

La metodología luego requiere de la generación de secuencias de video, lo cual es un proceso iterativo y a veces repetitivo. El objetivo es obtener un conjunto de muestras representativas de la variación de los parámetros seleccionados, el cual será posteriormente evaluado por el panel de observadores. Se obtiene entonces un conjunto de muestras de video resultantes de la alteración de los parámetros de red seleccionados a priori. Cada muestra de video es seleccionada porque contiene algún efecto de distorsión en la calidad. Se trata de muestras de unos 10s de duración, con su caracterización de GOPs.

Una vez obtenidas las muestras de video, se plantea la observación y puntuación de las secuencias distorsionadas. Esto se realiza en tiempo real, reproduciendo las muestras en una tira de video, como se explicará en el Capítulo IV. Notar que si bien se hace en tiempo real para el observador, es una tira de video procesada, de manera de que los valores de los parámetros representativos (pérdida de I, de P, de B, así

como de pérdidas de red y de ráfaga de pérdidas) se encuentren dentro de un rango razonable de variación.

Para la evaluación de los resultados obtenidos de la votación de los observadores es generalmente necesario hacer un filtrado de los resultados en un intervalo de confianza determinado (típicamente al 95% como recomienda la norma ITU-R BT.500-11). Para cada secuencia de video existirá un valor medio de calidad percibida y una desviación estándar.

Con los resultados filtrados se confeccionará una tabla general con las evaluaciones de calidad (ver tabla obtenida en Anexo 2) de cada secuencia, de la cual se seleccionarán los datos para obtener la fórmula de calidad PSQA, recurriendo a una RNN feed forward (realimentación hacia adelante, desde la capa de entrada hacia la(s) intermedia(s) y la de salida). El entrenamiento de la RNN se realizará a partir de la tabla antes mencionada, separando un 80% para entrenar y un 20% para validar la fórmula. En este trabajo se brindará una solución explícita con una RNN de 2 niveles y una solución más compleja y precisa, con 3 niveles.

### **III.4 Factores que inciden en la calidad**

Para cualquier evaluación formal de calidad, particularmente PSQA, es necesario contar con parámetros concretos, que sean medibles y permitan la obtención de resultados razonablemente objetivos a partir de realimentaciones subjetivas como las observaciones.

Una primera aproximación a estos parámetros puede ser la siguiente clasificación:

- 1) Parámetros ambientales: se refieren a las condiciones (óptimas) de evaluación del video, tales como luminancia ambiente y del video, contraste, distancia del observador a la pantalla, ángulo de visión, etc.
- 2) Parámetros de configuración inicial: en este rango se encuentran aspectos tales como la selección del códec (MPEG-2, MPEG-4, otros), la tasa y el esquema de codificación, el tamaño de *chunk*, el tamaño del *buffer activo* de cliente, entre otros.
- 3) Parámetros dinámicos: En este grupo se incluyen las pérdidas, ya sea a nivel de red (BER o similar) o de aplicación (pérdidas por tipo de cuadro), el tamaño de ráfaga de pérdidas, el retardo, o la variación de retardo (el *jitter*), etc.

Claramente los parámetros ambientales no son controlables (a menos que sea un laboratorio o condiciones de prueba), son manejados por el usuario, pero podemos asumir que en todo caso impactan de manera homogénea en cualquier flujo de video que el usuario observe mientras está conectado, independientemente de los otros factores de calidad.

El segundo grupo de parámetros sí es controlable en gran medida, siempre y cuando se aplique al funcionamiento en una red concreta en la que el servidor y el cliente están definidos y podemos seleccionar la codificación del flujo de video. No es recomendable en general la transcodificación del flujo de video, puesto que la calidad se degrada mucho en cada paso, pero si no hay otra opción tratar de elegir H.264.

La tasa y el esquema de codificación depende de la calidad objetivo deseada o el perfil de la aplicación buscada, es decir, si se pretende posicionar en un segmento de aplicaciones de alta calidad de video o en un segmento de baja a media calidad. El tamaño del *chunk* y el del *buffer activo* dependen de consideraciones del transporte a emplear, puesto que incide el retardo y el MTU medio de la red sobre la que se trabaje. Si bien hay redes inalámbricas cuyo retardo es muy sensible al tamaño de paquete, como GPRS, en general es posible fijar los valores de dichos parámetros a priori, de acuerdo al comportamiento promedio o caracterización de la red de transporte más típicamente utilizada.

Finalmente los parámetros dinámicos de calidad son aquellos que varían con el tiempo, según la aplicación y la red de transporte empleada. Para esto es necesario caracterizar la red de trabajo y el comportamiento del servidor del flujo de video (la fuente) de manera de posibilitar la reproducción de la realidad en una maqueta de pruebas. Como este trabajo se centrará en las variaciones de red, en la siguiente sección se caracterizan dos redes de acceso típicas: una red cableada tipo xDSL y luego una red de acceso celular de banda ancha como 3G UMTS o HSDPA.

### ***III.5 Caracterización de las redes de transporte***

Para evaluar la calidad de experiencia de video ante alteraciones de red, es necesario brindar parámetros que de alguna manera caractericen a las redes ADSL y 3G. En este trabajo se caracterizará a las redes de transporte por su ancho de banda (limitado a 512 kbps), el retardo, la variación del mismo (jitter), la pérdida y tamaño medio de ráfaga de pérdidas (MLBS) en el caso de una red celular. En ambos casos se fijan los parámetros antes de cada transmisión de flujo de video para simplificar las pruebas.

#### **III.5.2 Acceso xDSL**

Para este trabajo se ha considerado ADSL como el acceso de referencia, dado que es el más frecuente acceso en los hogares. Este acceso es bien conocido y no requiere de mayores detalles. De particular interés para la transmisión de flujos de video es el retardo, la variación del mismo y la tasa de pérdida de paquetes. Los valores típicos de retardo considerando resultados obtenidos para diferentes redes en [11] y [12] son hasta 20ms o 40ms medido como Round-Trip-Time (10 a 20ms considerando retardo de una vía). En cuanto a las tasas de pérdida de paquetes, usualmente las redes de los proveedores de servicio están diseñadas para baja pérdida, y lo máximo observado en [12] (y aceptable para MPEG-2) es de 5%, pero en este estudio se considerará un valor límite del 10% contemplando los rangos evaluados en [1] y para aproximarse un poco más al contexto de una red de acceso celular.

La variación de retardo o jitter es un parámetro que se debe mantener lo más controlado posible porque afecta sensiblemente los flujos en tiempo real como los de video. El control de jitter en una red de transporte es complejo, sobre todo considerando su acumulación en los diferentes saltos de la misma. No obstante puede ser compensado en cierto modo del lado del cliente por buffers activos, como incorpora GoalBit. Un valor de jitter generalmente aceptado en la industria de telecomunicaciones para el diseño de backbone multimedia es del 2% del valor del retardo, como es el caso del backbone implementado por Alcatel-Lucent [56] en la red celular de Antel Uruguay [87].

### III.5.3 Acceso celular

El acceso celular, como cualquier acceso inalámbrico, está fuertemente condicionado por la propagación en los diversos entornos. La variabilidad de las tasas de transmisión ascendente (UL) y descendente (DL) es bastante notoria en cualquier acceso de datos celular. Partiendo de GSM (*Global System for Mobile communications*) con su acceso de datos GPRS (*General Packet Radio Service*) o su evolución (*Enhanced Data Rates for GSM Evolution*) EDGE [82], hasta 3G y HSDPA han evolucionado la capacidad promedio de transferencia de datos en la red y los parámetros de calidad. Debido a que la aplicación de flujo de video de alta calidad como GoalBit demanda un ancho de banda de 384 a 512 kbps, el acceso que cubre esos requerimientos es 3G y en mejor manera HSDPA.

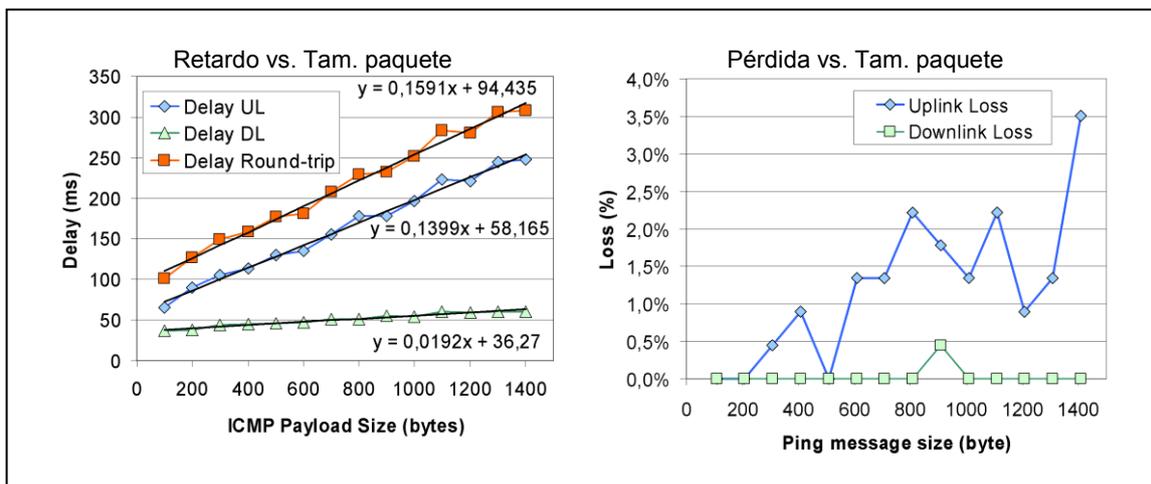
Cabe señalar que 3G, al ser una red de tecnología de acceso basada en multiplexación de códigos, la carga de usuarios de la red tiene un gran impacto sobre la tasa de transferencia alcanzable. Es por eso que en 3G es necesario un control muy fino de la potencia de transmisión y del sincronismo, existiendo factores de carga máximos por cada celda de 3G (mínima área de cobertura inalámbrica de 3G), excedidos los cuales las radiobases de 3G rechazan nuevos usuarios.

Otra consideración importante de 3G es que clasifica los flujos de datos en lo que se denominan *Radio Access Bearers (RAB)* o portadoras de radio, que presentan un conjunto de ajustes de parámetros de calidad, pero el más relevante es la tasa de transferencia. Es así que tenemos RAB de 256, 384 o 512 Kbps por ejemplo (la disponibilidad de estas tasas puede variar según el proveedor de equipos), por lo que las tasas de transferencia en una red 3G están escalonadas. Cuando en este acceso se plantea transmisión de video se utilizan RAB de 384 o 512 Kbps, claro que el tamaño de pantalla condiciona esta tasa también (varía la tasa de codificación), o sea que si se plantea para una laptop en lugar de un celular, la tasa de 512 Kbps es más adecuada. Adicionalmente 3G maneja clases de servicio: *Background, Interactive, Streaming, Conversational*; lo que permite cumplir con requerimientos de calidad específicos de los flujos de video en vivo como es el retardo y variación del mismo.

Con HSDPA se pueden obtener teóricamente tasas de hasta 14 Mbps, aunque la mayoría de las redes existentes soportan (por limitaciones de la tecnología de acceso) velocidades máximas de 3,6 o 7,2 Mbps y los terminales de usuario más comunes (como módems GPRS/EDGE/3G/HSDPA) soportan típicamente 1,8 Mbps. Conceptualmente HSDPA está pensada para mejorar las tasas de DL de la red celular, que es el que generalmente demanda mayor velocidad. En el enlace ascendente,

HSDPA generalmente comparte recursos con 3G, utilizando RAB de 128 Kbps típicamente. También es posible utilizar el nuevo canal dedicado definido en HSDPA, mejorando la velocidad y retardo del enlace ascendente, pero en general esto se hace una vez madurada la red HSDPA del operador porque es más costoso. Otra fórmula para mejorar la asimetría de HSDPA es introducir HSUPA (equivalente a HSDPA para el UL, llegando a tasas del orden de 3 Mbps), pero es una tecnología aún en estado incipiente en las redes celulares y en los equipos de acceso del mercado. No obstante esta asimetría, HSDPA presenta un retardo promediamente menor que 3G al reducir el TTI (intervalo entre transmisiones) de un mínimo de 10 ms en 3G a 2 ms.

En este trabajo se considerará una red de acceso 3G/HSDPA como modelo, al ser una combinación de vanguardia que presenta mejor capacidad para un sistema de distribución de video en vivo como GoalBit. En [5] se realiza un estudio del acceso celular 3G/HSDPA que es interesante por presentar mediciones del retardo de la red y de la tasa de pérdidas. La figura 3.5 muestra los resultados obtenidos de retardo y pérdida en función del tamaño de paquete.



- Figura 3.5: Retardo y pérdida en función del tamaño de mensaje (ICMP) en una red 3G/HSDPA según [5] -

Se puede observar que efectivamente la asimetría en el UL produce mayores retardos (se desprecia aquí el comportamiento “escalonado” que se observa en la realidad para la dependencia entre retardo y tamaño de paquete). La variación del retardo será mayor también en el UL que en el DL.

Para simplificar las pruebas los parámetros de red no se variarán durante la transmisión de un flujo de video (sí para diferentes repeticiones). Esto es aceptable si se consideran dependientes del tamaño de paquete, que es fijado por el tamaño del *chunk*. Como el *chunk* en GoalBit en general es mayor que el MTU, el mismo será fragmentado al tamaño de la MTU, lo que será del orden de 1400 bytes. Adicionalmente, dada la asimetría UL/DL, es necesario diferenciar los parámetros que se ajustarán en cada sentido de la red. Para los 1400 bytes, el retardo UL/DL es de 268/64 y la pérdida de paquetes UL/DL es de 3,5/0,5 %. Para la variación de retardo puede considerarse la recomendación de diseño de Ericsson [57] en [88], de entre 2 y 5ms (2-10% del retardo).

El tamaño medio de ráfaga de pérdidas en una red celular 3G/UMTS puede considerarse variable entre 1 y 10 paquetes consecutivos perdidos, como en [1] o en [19] que se considera un rango de de 3 a 9. No obstante en este trabajo se plantea una densidad de pérdida nunca superior a 4, ya que es la tasa máxima de pérdida encontrada en una red celular de acuerdo a [5].

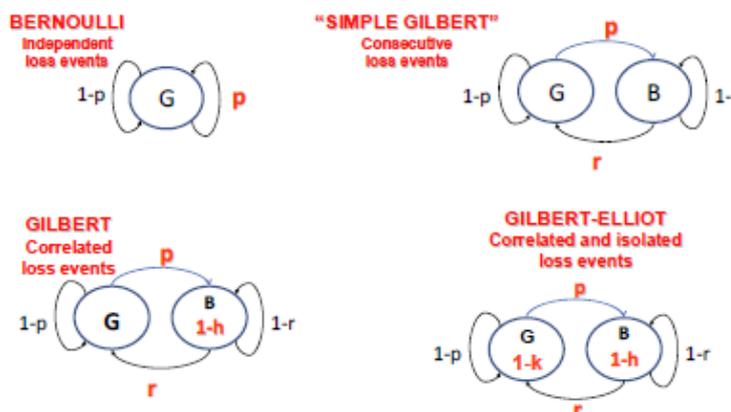
Otro estudio interesante de QoE en redes celulares HSDPA y aplicando PSQA es el presentado en [99], donde incluso se propone un esquema de control de congestión en el acceso para mejorar la performace de aplicaciones como la de *Video Streaming*. El estudio, si bien trabaja con PSQA como herramienta de evaluación, se concentra en definir una configuración parámetros de calidad de la red 3G/HSDPA que permita garantizar (en un intervalo de confianza al 95%) una QoE mínima para los usuarios.

### III.5.4 Herramienta de modelado de la red

Para modelar la red se trabajará con Netem [6], o más precisamente su extensión Netem2 [13]. La ventaja de Netem es que es un conocido emulador de red y que integrado al núcleo de Linux.

Netem modela bien el retardo, el jitter (la precisión del tic de reloj de Linux, que es la que condiciona la precisión de Netem es razonable) y la pérdida aislada, pero no modela el MLBS. Para ello es necesario recurrir a un modelo más avanzado como incorpora Netem2. Netem2 es propuesto en [13] respondiendo ante las falencias de Netem a la hora de modelar una red real. Incorpora modelos reconocidos ampliamente en la literatura, desde el de Bernoulli (1 estado), el de Gilbert [104, 105] (2 estados) en su versión simplificada y normal y el de Gilbert-Elliot. En particular en [89] Netem y el modelo de Gilbert son usados para emular una red UMTS.

En la figura 3.6 se muestran los 4 modelos mencionados en [13], donde G representa el estado “bueno” y “B” el estado “malo” o de pérdida. Netem sigue aproximadamente el modelo de Bernoulli, con un único estado en el que pierde paquetes con cierta probabilidad ( $p$ ). Esto manifiesta su incapacidad de modelar pérdidas consecutivas correlacionadas, como las ráfagas.



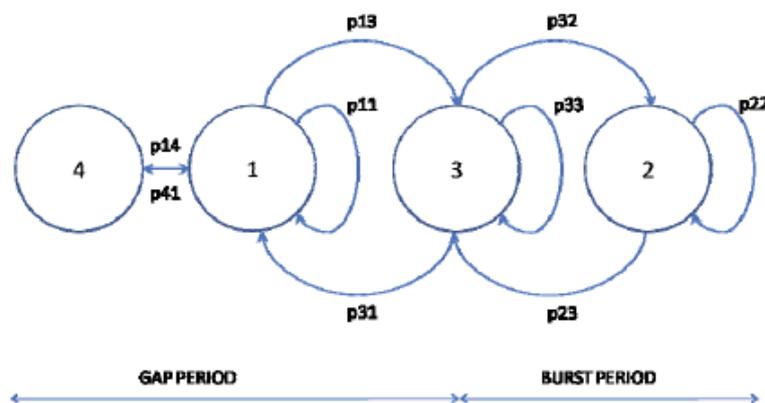
- Figura 3.6: Modelos de Markov contemplados en la propuesta de Netem2 para la emulación de parámetros de red (pérdida, jitter, retardo, ráfagas, etc). Ver [13] -

El modelo de Gilbert simplificado es el generalmente más adoptado por su simplicidad de implementación agregando una sustancial mejora a la aproximación de una red real como son las ráfagas de pérdidas. Se define entonces un estado malo en el que pueden perderse varios paquetes (eventos de pérdida consecutivos).

El modelo de Gilbert tiene también dos estados pero 3 parámetros, porque en estado “malo” aún existe una posibilidad ( $h$ ) de transmitir correctamente un paquete. Se podría decir que existe una “densidad” de ráfaga de pérdidas.

El modelo de Gilbert Elliot por su parte tiene 4 parámetros independientes. Lo que se intenta modelar es un sistema en el que si bien se puede diferenciar un estado “bueno” del “malo”, existe una cierta (aunque baja) probabilidad de pérdida ( $1-k$ ) también en el estado bueno, en el cual se pueden considerar pérdidas aisladas.

La propuesta de Netem2, reconociendo que son todos en definitiva modelos de Markov de estados múltiples, es integrarlos en lo que denomina un modelo General e Intuitivo (GI), como se muestra en la figura 3.7.



- Figura 3.7: Modelos General e Intuitivo de Netem2 –

En [13] se relacionan los parámetros entre el modelo propuesto y los 4 modelos clásicos antes mencionados con un mapeo intuitivo de los parámetros de los modelos al comportamiento observado en la realidad.

En particular en este trabajo se usará Netem2 con la aproximación de 2 estados como el de Gilbert Simplificado, ampliamente referenciado en [1, 14, 78], incluso aplicado a transmisión de video sobre una red celular. En este modelo, los únicos parámetros que permanecen son  $P_{Loss}$  y MLBS o  $E(B)$  y el funcionamiento del sistema se centra en el estado 3, que se pueden reflejar como:

$$\begin{cases} P_{31} = \frac{1}{E(B)} \\ P_{13} = \frac{P_{Loss}}{E(B) \cdot (1 - P_{Loss})} \end{cases}$$

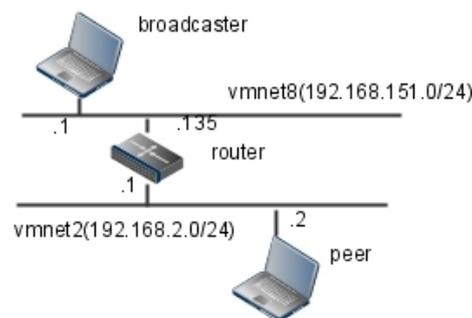
Siendo  $P_{31} = r$  y  $P_{13} = p$  del modelo original simplificado de Gilbert.

## IV Aplicación de la metodología PSQA

En este Capítulo se describen los pasos seguidos para la aplicación de la metodología PSQA a la plataforma GoalBit, profundizando los pasos generales descritos en III.3, comenzando con la implementación de la red de pruebas. El objetivo es generar un ambiente de pruebas que permita reproducir las condiciones observadas en las redes de transporte ADSL y 3G como se detallaron en III.5 y así medir el impacto de los parámetros de red en la calidad de video para finalmente obtener una función genérica que mida la calidad de video.

### IV.1 Configuración de la red de prueba con GoalBit

Para evaluar la red de transporte se utilizó una computadora con capacidad de enrutamiento (*router* en la figura), aunque es suficiente con que permita *packet forwarding*. Sobre dicho nodo se instaló la aplicación NetEm y luego NetEm2 (modificando el Kernel), que son paquetes de software concebidos para alterar parámetros de red como retardo, jitter, pérdidas y ráfagas de pérdidas. El uso de Netem2 fue necesario específicamente para emular ráfagas de pérdidas. Por tiempo y flexibilidad de cambios (se debió aplicar un kernel específico y recompilarlo con el nuevo módulo según detalles en Anexo 1), se optó por utilizar una máquina virtual para *router*. Luego se configuraron separadamente un servidor y un cliente GoalBit de manera de independizar las configuraciones de ambos nodos, mejorar performance y desacoplando eventuales problemas (además presenta mayor claridad).



- Figura 4.1: Red de pruebas con GoalBit para la generación de secuencias –

### IV.2 Descripción general de la metodología

La metodología de pruebas planteada en este trabajo tiene tres etapas clave: la primera es reproducir en la plataforma de pruebas las condiciones de una red de transporte ADSL y 3G, de modo de obtener muestras representativas en todo el espacio de variación de parámetros de ambas redes. La segunda es traducir la variación de los parámetros de red a parámetros más abstractos y extensibles como son las pérdidas por cuadros de video. Y finalmente en la tercera etapa se debe armar

una tabla con el conjunto de muestras de video con los parámetros de entrada seleccionados y la calidad observada como salida.

La primera etapa es configurar la maqueta descrita en la sección precedente y transmitir diferentes flujos de video por la misma alterando los parámetros de retardo, jitter, pérdidas y ráfagas de pérdidas. Se trata de observar el impacto cualitativo en diferentes fuentes de video, seleccionando videos de características de movimiento, color y foco de imagen. En este trabajo se seleccionaron 5 fuentes de video de una duración entre 3 y 5 minutos, normalizando las mismas en tamaño y tasa de codificación de video, la cual se fijó en 384 kbps +/- 20% para ser compatible con el ancho de banda. Notar que la misma tasa sólo que con un margen del 10% se propone en [99] para evaluar la transmisión de video sobre redes HSDPA.

En la segunda etapa se busca obtener una relación entre los parámetros a nivel de red y los parámetros a nivel de aplicación. Se hacen variar los parámetros de red dentro del rango dado en III.5 mientras se transmiten los flujos de video de algunos minutos de duración y diferentes fuentes. Observar que la elección de diversas fuentes es necesaria para independizarse de características particulares de un video fuente. Si bien la tasa de codificación de video se promedió en 384kbps +/- 20% para mantener los máximos dentro del ancho de banda seleccionado y se ajustaron todos los videos fuente al mismo tamaño en pixeles, la tasa de video instantánea, el contenedor más adecuado para la transmisión y parámetros tales como la cantidad de marcos de cada tipo o el movimiento de cada fuente son diferentes.

Durante las pruebas de esta etapa se pretende investigar la incidencia de los diferentes parámetros planteados. Variando en principio sólo el retardo y el jitter, se observa que en los rangos manejados, incluso para valores de retardo de 350ms 5ms de jitter como se ven en las redes 3G, son absorbidos por los buffers de la aplicación, por lo que el impacto introducido es mínimo. Nótese además que típicamente los buffers de aplicación se diseñan para la red de transporte concreta y con holgura suficiente. Por lo tanto son las pérdidas y ráfagas de pérdidas (con su densidad correspondiente) los parámetros de red más relevantes. No obstante trabajar a nivel de parámetros de red, si bien es simple de medir, es fuertemente dependiente de la red de transporte en cuestión. Si se logra un mapeo entre parámetros de red y parámetros a nivel de aplicación, éstos últimos serían mucho más extensibles y apropiados para diversos contextos de red como fallas en la red de transporte o ante desconexiones de peers por ejemplo. Parámetros análogos a nivel de aplicación serían pérdidas y ráfagas de pérdidas por tipo de cuadro de video. La dificultad reside en encontrar la relación entre estos parámetros a nivel de red y a nivel de aplicación.

No se trata de una traducción lineal o evidente siquiera, ya que los cuadros se transportan en contenedores formando *chunks* de tamaño fijo y luego los buffers y las retransmisiones realizadas por la aplicación introducen alinealidades. Es por eso que la relación se obtendrá empíricamente: fijando el retardo en un valor medio de 200ms y el jitter en un 2%, se hacen variar las pérdidas y ráfagas en todo el rango indicado en III.5 y luego se estiman las pérdidas y ráfagas de pérdidas por tipo de marco, obteniendo un rango de variación de las mismas.

Para obtener las pérdidas y MLBS por cuadro se debe estimar la probabilidad de pérdida para el flujo de video. Dado que lo que se va a obtener de esta etapa,

siguiendo la metodología SS del estándar ITU-R BT.500-11, son secuencias de video de 10 seg de duración resultantes de la variación de los parámetros de entrada, la estimación de probabilidad de pérdida se realiza a partir de dichas muestras.

Para esto se calcula la pérdida por tipo de cuadro como el cociente entre cuadros recibidos y enviados en la correspondiente ventana de 10s. Si se considera cada una de estas secuencias de video como un “muestreo discreto” de una variable estadística que es el flujo de video y se toma un grupo de secuencias de 10 seg de video dentro del flujo (al menos 10), se puede calcular un estimador de la media de pérdida como la media de las pérdidas en cada secuencia de 10s.

Para obtener los parámetros de entrada de la función PSQA se consideran las pérdidas correspondientes a la secuencia (de las 10 o más consideradas) que presente menor desviación respecto a la media. Notar que la media puede no ser una secuencia del grupo y por ende no se podría reproducir a un espectador, por eso se toma la de menor desviación presente en el grupo.

Observar que este procedimiento se debe repetir varias veces para cada flujo de video fuente (de los 5 elegidos) en las distintas configuraciones de parámetros de red, de manera de obtener un conjunto de muestras que cubra el rango de variación de pérdidas por cuadro aceptable, como se explicará en V.

En la tercera etapa se debe confeccionar una tabla que contenga los parámetros de entrada y salida para el conjunto de muestras de video de 10 seg seleccionadas. Con el conjunto de muestras se construirá una tira continua de video, intercalando transiciones en color sólido, como sugiere la metodología SS del estándar ITU-R BT.500-11. Esto permite reproducir el conjunto de muestras a los observadores como un flujo continuo, manteniendo la atención sin pausas ni repeticiones. El promedio de las evaluaciones de cada muestra entre los observadores es la salida de calidad de experiencia de cada muestra.

Una vez completada la tabla, se selecciona un subconjunto de filas de entrenamiento y otro de prueba de la RNN, en una relación 80/20 %. Se busca obtener una RNN de 2 niveles (sin capa intermedia) que posibilite expresar la salida como una función polinómica (ver V.I) y otra de 3 niveles. En ambos casos la RNN se determina en base al mínimo error cuadrático medio.

Antes de pasar a la descripción paso a paso del procedimiento, cabe señalar algunos aspectos relevantes de la prueba respecto a los “videos fuente” del flujo de video en formato H.264/MPEG4-AVC (denominados como video\_original en este procedimiento):

- *Diversidad de videos fuente:* Es conveniente tener diferentes videos fuente (alrededor de 5 videos), para que la selección de secuencias de muestra no sea monótona para el observador y sea rica en diversidad de características de imagen.
- *Tasa de codificación:* Los videos que se obtengan pueden tener una tasa de codificación variable que genera problemas en un streaming. Es conveniente transcodificar el video (el sonido no es relevante) a una tasa promedio aceptable para el ancho de banda de la red de transporte. En este trabajo se transcodificó a 384 Kbps promedio, con variaciones de hasta un 20%.

- *Tipos de cuadros:* Los videos tiene que ser variados en tipos de cuadros para medir el impacto de la red sobre todos los tipos de cuadros. En particular al transcodificar se debe cuidar la diversidad y proporción de cuadros a la salida (en Avidemux [109] con x264 [110] es fácilmente configurable)
- *Tamaño de imagen:* Desde un punto de vista procedural particular de este trabajo, dado que se opta por concatenar todas las secuencias de muestra de 10s, es fundamental respetar el mismo tamaño de imagen (evidente pero no por eso menos importante).
- *Tamaño de chunk:* La relación entre el tamaño de chunk y el de los VOP es importante a la hora de ver el impacto de las pérdidas de red. Cuantos más cuadros por chunk, mayor es el impacto de la pérdida de un paquete, por ende esta relación se debe cuidar. Es aceptable considerar hasta 5 cuadros por chunk. Si bien la selección de este parámetro está dentro de lo que se consideró un parámetro de configuración inicial, es conveniente revisarlo a la hora de las pruebas, si no el impacto de las pérdidas puede ser demasiado grande.

### IV.3 Procedimiento paso a paso

Un objetivo de este trabajo es permitir la reproducción de la metodología a futuro, lo cual motiva la inclusión de esta sección.

Diferentes opciones fueron probadas para la implementación de una maqueta de pruebas capaz de emular correctamente una red real. Cabe aclarar que el manejo de Linux en bajo nivel, de VLC [113] como aplicación de video base para GoalBit y de las secuencias de video no es trivial, por lo que involucró un tiempo subestimado en principio.

Se partió de un Linux con distribución Ubuntu 8.04, el cual debió ser actualizado a la versión 8.10. Sobre este punto cabe señalar que a menos que ya se tenga 8.04 instalado y actualizado, conviene actualizar directamente de 8.04 a 8.10, de lo contrario algunas de las librerías sugeridas por Ubuntu generan problemas de funcionamiento del video de GoalBit. Considerando esto, la instalación sigue la wiki [18].

Durante la etapa de pruebas y conocimiento de GoalBit se utilizó GoalBit 0.4, 0.5, 0.6 y 0.7. La transición a GoalBit 0.5 fue necesaria para actualizar librerías de VLC y algunos módulos requeridos para este trabajo (como el *pattern\_generation\_duplicate* y el *pattern\_dropper\_duplicate*). La transición a GoalBit 0.6 fue para probar el *Broadcaster* con la configuración más actualizada de librerías de VLC. Y GoalBit 0.7 fue necesario para variar el tamaño del chunk.

#### 1) Configuración de la red de transporte

Para emular las redes de transporte ADSL y 3G se recurre a NetEm y NetEm2 (recordar que este último es más completo porque permite emular ráfagas de

pérdidas). Ya sea durante la evaluación cualitativa de la etapa 1 antes mencionada, como para la determinación empírica de los parámetros de entrada de la función PSQA como se indicó en la etapa 2, se deberán transmitir muchos flujos por esta red emulada.

Se recomienda correr NetEm en un enrutador o puente separado para evitar camuflar las pérdidas de red con retransmisiones de TCP. En el caso de ser una máquina virtual, funciona más estable corriendo NetEm en la interfaz de la red privada o interna en lugar de la interfaz con el host. Se puede recurrir a NetEm en el servidor GoalBit (como se hace en este trabajo) para controlar el desbalance de retardo en un sentido necesario para el UL de la red celular.

A modo de ejemplo, para cada transmisión de video se puede configurar la red de la siguiente manera:

- a) Controlar bit rate: se define un flujo de clase TBF en NetEm en el servidor GoalBit sobre la interfaz en la que corre el *broadcaster* (eth0 en este caso). Si bien se agrega algo de latencia, el retardo se controlará en el router.

```
# tc qdisc add dev eth0 root handle 1: tbf rate 512kbit burst 256k latency 10ms
```

- b) Control de retardo, pérdida y MLBS: usando NetEm2 en el router se define en este caso un retardo de 50ms, un jitter de 2ms, una pérdida de 5 y una ráfaga de pérdidas de 3 paquetes.

```
# /usr/sbin/tc qdisc add dev eth0 root netem2 delay 50 2 loss_GI 5 3
```

- c) Para agregar retardo unidireccional, de UL celular: Es posible hacerlo sobre el tráfico entrante a un nodo se debe recurrir a una pseudo-interfaz IFB. Para eso se carga el módulo IFB y se levanta la sub-interfaz IFB0

```
# modprobe ifb
# ip link set dev ifb0 up
```

Y se ejecuta el filtro de NetEm en el servidor GoalBit sobre la interfaz en la que corre el *broadcaster* (eth0 en este caso)

```
# tc qdisc add dev eth0 ingress
# tc filter add dev eth0 parent ffff: \
  protocol ip u32 match u32 0 0 flowid 1:1 action mirred \
  egress redirect dev ifb0
# tc qdisc add dev ifb0 root netem delay 150ms
```

Todos estos ajustes son parte de un proceso iterativo, por lo que los parámetros se variarán de acuerdo a la necesidad. Finalmente, la regla se puede eliminar en cualquier momento sobre la interfaz afectada:

```
# tc filter del dev eth0
# tc qdisc del dev eth0 root
```

## 2) Preparación de GoalBit.

Una vez configurada la red de transporte, lo siguiente es transmitir el flujo de video entre servidor y cliente GoalBit.

### 2.a) Preparación en el lado servidor

Para el lado servidor se deben levantar el *Broadcaster* y el *Tracker*.

Para levantar el *Tracker*:

```
$ cd ~/svn/goalbit/src/opentracker/btv-opentracker/  
$ ./opentracker -V
```

Para levantar el *broadcaster*:

Es en general conveniente purgar las configuraciones de corridas previas de GoalBit. Adicionalmente, si en algún momento da problemas al levantar la configuración global del broadcaster, borrar la copia temporal del archivo.

```
$ cd ~/svn/goalbit/src/trunk  
$ rm -rf ~/.local/share/goalbit/  
$ rm -rf ~/.config/goalbit/
```

El video a transmitir por la red en este trabajo se transcodió a 384 Kbps promedio usando con Avidemux: AVC (librería x264) para el video y AAC (FAAC) para el audio, con multiplexor ASF. De esta forma se logra una codificación más robusta para transmitir en 512 Kbps, especialmente sobre redes celulares. Esto se logra con la siguiente secuencia de comandos:

```
$ ./goalbit -vvv -I broadcaster_manager \  
--broadcaster-muxer "asf" \  
--broadcaster-conf ~/svn/p2ptv/goalbit/tests/lan-adaptor/broadcaster.conf \  
--broadcaster-output ~/svn/p2ptv/goalbit/tests/lan-adaptor/test.goalbit \  
--broadcaster-input ~/Desktop/video_1.asf
```

Donde,

- *Broadcaster-muxer* indica el contenedor a usar (en general ts es el que mejor funciona para la obtención de las GOPs en el *peer*)
- *Broadcaster-conf*: es el camino al archivo de configuración, que indica el número de broadcasters a levantar, la IP y el puerto donde corre el broadcaster y el ancho de banda de subida y bajada

```
$ cat broadcaster.conf  
  
1  
192.168.151.1,6900,512,512,4
```

- **Broadcaster output:** es el archivo `.goalbit` que verá el *peer* como referencia al video.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<channel_list size="1" active_channel="test_channel" >
  <channel id="test_channel">
    <chunk_size>65536</chunk_size>
    <tracker_url>http://192.168.151.1:6969/announce</tracker_url>
    <bitrate>512</bitrate>
    <name>Test Channel</name>
    <description>Canal de testing</description>
    <thumb>http://www.adinettv.com.uy/dynamic-
data/canales/logo_liguilla.gif</thumb>
  </channel>
</channel_list>
```

Antes de pasar al lado cliente, es necesario generar para cada video a transmitir por la red la estructura en tipos de cuadros o GOPs original del mismo. Para ello se utiliza el módulo de GoalBit *pattern\_generation\_duplicate*, que nos permite generar un archivo con los cuadros de video (GOP) leídos. A su vez se re-direcciona la salida a pantalla también para asegurarnos de mantener el sincronismo en el video y la escritura correcta de las GOP en el archivo `gops_tx_sinp.txt`.

```
$ ./goalbit -I dummy \
--sout="#pattern_generation_duplicate{dst=display, dst=std{access=file, \
mux=asf, dst=/tmp/video.asf}, pattern ~/gops_tx_sinp.txt}" \
~/Desktop/video_1.asf
```

## 2.b) Preparación en el lado cliente (peer)

En el *peer* correremos GoalBit usando dos módulos auxiliares: el *adaptor*, que nos permite manipular y re-direccionar la salida del flujo de video y el *pattern\_generation\_duplicate*, al igual que en el servidor, para generar el archivo de los cuadros de video recibidos, que denominamos `gops_rx_conp.txt` en el siguiente ejemplo.

```
$ ./goalbit -vvv -I output_adaptor \
--bt-conf ~/svn/p2ptv/goalbit/tests/lan-adaptor/peer.conf \
--adaptor-output="pattern_generation_duplicate{dst=display, \
dst=std{access=file, mux=asf, dst=/tmp/video.asf}, \
pattern ~/gops_rx_conp.txt}"
```

Si bien se trata de capturar el inicio de la secuencia para aprovechar todo el video, un pequeño retardo en la captura en el *peer* responde al comportamiento normal mientras un *peer* se "sincroniza" con la red P2P, detecta el primer contenedor de video.

### 3) Estimación de pérdidas

Para realizar lo que corresponde a la etapa 2 de la metodología, se deben determinar los parámetros de entrada de PSQA, que son como se explicó en la sección anterior, las pérdidas y ráfagas de por tipo de cuadro.

- a) El primer paso para estimar las pérdidas es obtener es un archivo combinado de GOPs originales y GOPs recibidas luego de pasar por la red de transporte, de manera de identificar los cuadros perdidos. Para ello utilizamos el script *merge.sh* (se asume que los archivos están en el mismo directorio que el script).

```
$ ./merge.sh
Ingrese el nombre del archivo de GOPS original (sin pérdidas):
gops_tx_sinp.txt
Ingrese el nombre del archivo de GOPS del peer (con pérdidas):
gops_rx_conp.txt
Ingrese el nombre del archivo de GOPs combinado [merged.csv]:
merged_ejemplo.csv
```

El archivo de salida (por defecto merged.csv) devuelve en formato csv, e indica un cuadro perdido con un 0 en el lugar de cuadro leído y agregando “missing frame” como sufijo al md5. Lo importante es el “0” para proceso posterior.

```
P,0,2a860437cedda28d66b7aa1524a300d8 – missing frame
B,1,2347,2b860586cedda28d66b7aa1524a259c7
```

- b) El segundo paso es obtener las secuencias de 10s requeridas por la metodología SS. Lo que se hace es simular *peers* que se conectan a la red en diferentes momentos (selección aleatoria de inicio de captura) usando el script *secuencias.awk*. Este script toma el archivo merged.csv y devuelve 2 tipos de archivos. Por un lado un resumen de las secuencias (en el ejemplo, ghost\_probs), el cual será usado para estimar la probabilidad de pérdida por cuadro y por otro los archivos de GOP correspondientes a cada una de las N secuencias, los cuales se deben conservar hasta finalizar el paso 3c.

```
$ ./secuencias.awk merged/merged_ghost_10_1.csv

Ingrese prefijo que referencie el video fuente: ghost

$ head secuencias/ghost_probs

196,8279c71560177ec82fc16de431fc4f31,496,0,0.0178,0.0278,0,1,1

763,6466e54d6db5c9cb5d4bd5d62165db17,1063,0.1333,0.0050,0.0345,1,1,1

2540,3036ed481c17dd6376f87a8fe72874a1,2840,0,0.0892,0.0263,0,1,1
```

Los archivos de GOPs están en el formato requerido por el módulo de GoalBit *pattern\_dropper\_duplicate* para reproducir el flujo de video con pérdidas. Se nombra como *<video>\_secuencia\_<ind\_relativo>\_<ind\_abs>*, ej. *ghost\_secuencia\_2305\_3022*.

- c) Finalmente para seleccionar una secuencia representativa del impacto de red como se explicó en la etapa 2 de la descripción general de la metodología, se utiliza el script *prob\_orig.awk*. Seleccionando la opción 2 calcula la media muestral de las N secuencias de 10 segundos obtenidas, con las respectivas pérdidas y MLBS, a la vez que indica la secuencia del grupo que más se acerca a la media muestral.

```
$ ./prob_orig.awk secuencias/ghost_probs
Indique archivo de partida para obtener probabilidad: merged completo (1)
o de las secuencias (2)
2
Prob_perdida(I)= 0.0855      desv= 0.1826      MLBS I= 0.3375
Prob_perdida(P)= 0.0840      desv= 0.1926      MLBS P= 1.3767
Prob_perdida(B)= 0.0751      desv= 0.1529      MLBS B= 1.0083

la secuencia seleccionada es:
pos: 2-> 2540,3036ed481c17dd6376f87a8fe72874a1,2840,0,0.0892,0.0263,0,1,1
```

#### 4) Construcción de la tabla y tira de video

Con las muestras obtenidas en 3 se construirá una tabla de entradas para cada muestra de video y la tira de video a reproducir a los observadores.

Para armar la tira de video a reproducir a los observadores, se recurre a la capacidad de GoalBit de reproducir un video distorsionado a partir del video original y un archivo de GOPs con pérdidas. Por lo tanto para esta parte será necesario construir separadamente la tira de video del archivo de GOPs con pérdidas correspondientes. El procedimiento para esto es el siguiente:

- a) Obtención del archivo de GOPs con perdidas: Tomando el archivo de GOPs con pérdidas obtenido en 3b correspondiente a la secuencia seleccionada en 3c, se tiene el archivo de GOPs con pérdidas de la secuencia. Luego se debe obtener el archivo de GOP correspondiente a la transición (que no tiene pérdidas), como se indicó en 2a. Repitiendo este procedimiento para cada muestra seleccionada, se obtendrá el archivo de GOPs con pérdidas para toda la tira de video.
- b) Obtención de la tira de video: para obtener la tira de video se puede utilizar Avidemux, cortando según el índice absoluto de la(s) secuencia(s) seleccionada(s) en 3c (por ej. *video\_1.asf*). Es conveniente cortar 2 o 3 cuadros antes del cuadro I indicado por el índice absoluto, para darle tiempo al reproductor de sincronizarse con el cuadro I. Este procedimiento se repetirá

para cada muestra seleccionada, intercalando entre ellas la transición en color sólido.

Cabe señalar que debido a la naturaleza de la generación del archivo de GOPs a partir del reproductor, es conveniente comparar cada tanto (por ejemplo cada 10 muestras concatenadas) la secuencia de GOPs obtenida a partir de la tira (como se indica en 2a) con el archivo de GOPs con pérdidas construido. Es posible que se deban agregar algunos cuadros faltantes en el archivo de GOPs con pérdidas (debido a que las muestras se tomaron cortando en un cuadro I).

## 5) Evaluación de secuencias

En este trabajo se seleccionaron 106 videos en total, incluyendo muestras de los videos originales sin distorsionar, a modo de referencia de calidad para el observador. El promedio de las evaluaciones en los observadores es la calidad de experiencia, salida de la función PSQA necesaria para completar la tabla de entrada elaborada en 4. La tabla construida con los diferentes parámetros contemplados y la calidad resultante se presenta en el Anexo 2.

- a) Se trabaja con un grupo de 10 personas, a las cuales se les muestran las secuencias con pérdidas, intercaladas con alguna secuencia sin pérdida a efectos de referencia.
- b) Cada tanto se repite alguna secuencia de un video diferente pero con igual tasa de pérdida para mejorar la evaluación.
- c) Cada observador deberá evaluar el video en una escala de 1 a 5.
- d) Se promedian las puntuaciones de cada secuencia para obtener una media de calidad para las diferentes combinaciones de entrada de parámetros. Es posible descartar aquellas evaluaciones que queden fuera de un intervalo de confianza del 95%.

## 6) Obtención de la función PSQA con RNN

La etapa final de este proceso es el entrenamiento y validación de la RNN.

Para generar la RNN de tres niveles se debe determinar la cantidad de neuronas de la capa intermedia. Esto se hace iterando desde 1 neurona, calculando el error cuadrático medio resultante en la validación (repetiendo pasos 6a a 6c) y terminando en la cantidad de neuronas que logra el menor error. En este trabajo se definieron 3 neuronas en la capa intermedia.

Los pasos son los siguientes:

- a) Se genera primero el formato de la red neuronal, que en este caso es de 3 capas con 4 neuronas de entrada, 8 totales y una de salida. Esto genera un archivo de salida *rnn\_gen\_con1.dat* con el modelo de la RNN

```
$ ./ff_gen_con 4 8 1
```

- b) A partir de la tabla de evaluación de secuencias obtenida en 5, se conforman los archivos de entrenamiento y de verificación de la RNN. Dichos archivos deben tener el siguiente formato:

```
163 # Número de filas de entrenamiento

#entrada1  entrada2  entrada3...
1.000000  0.970000  0.808000
0.000000  0.000000  0.000000
...
#salida (evaluación de calidad normalizada 0-1)
0.920000
0.000000
```

- c) Entrenar y verificar la RNN. Notar que es un proceso iterativo tratando de minimizar el error. En general requiere normalizar e invertir la entrada de manera que sea monótona creciente entre 0 y 1 (por ejemplo 1 – “pérdida”, como se comenta en VI.1). Eventualmente también puede mejorar la aproximación el agregado de ciertos puntos conocidos forzados, como (0,0) y (1,1).

```
$ ./ff_rnn S rnn_gen_trn1.dat # entreno
$ ./ff_rnn T rnn_gen_tst1.dat # verifico
```

- d) Generación de gráficas. El script toma un archivo .dat de entrada con los datos a graficar, con el siguiente formato:

```
plot_LRi-LRp.mat # nombre archivo salida .mat:
0 1 0.05          # eje X (orig fin intervalo):
0 1 0.05          # eje Y (orig fin intervalo):
# Neuronas a graficar (segun archivo WTS):
0 1              # Neuronas de entrada, lambda (X Y):
3                # Neuronas de salida, o (Z):
.02 .11 .99      # Un punto medio:
1 1 10           # Escalado X Y Z:
PSQA             # Título:
LRi              # Etiqueta X:
LRp              # Etiqueta Y:
Q                # Etiqueta Z:
```

Luego se genera el archivo .mat, el cual se levanta con Matlab [108] para graficar y exportar la gráfica en pdf.

```
$ ./generate_3Dmats ./plot/plot_LRi-LRp.dat
```

#### IV.4 Observaciones sobre el proceso

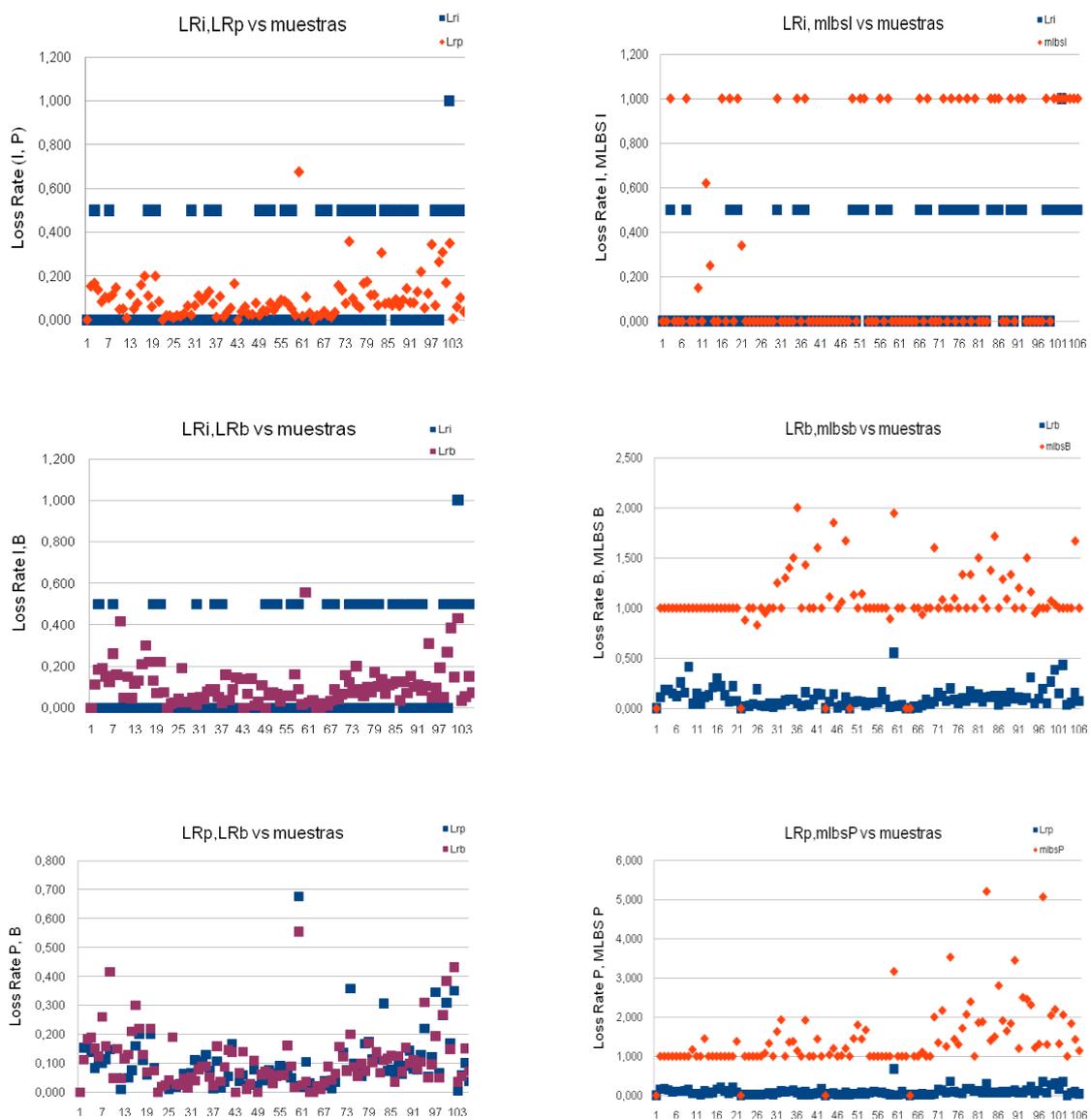
Resultan pertinentes algunas observaciones sobre el trabajo con flujos de video y la plataforma de pruebas:

- Los factores contribuyentes a la calidad del video se entremezclan: es común observar pixelados focalizados o por zonas, congelamientos de imagen, o incluso desfasajes horizontales o verticales del cuadro de imagen, combinados o por separado. Lo complejo es aislar la causa, ya que puede ser ancho de banda, o pérdidas, o problemas en el *broadcaster* o el cliente. Incluso dentro del cliente puede ser causado por el propio reproductor o por el sistema operativo (problemas de memoria que afectan al reproductor o problemas con las máquinas virtuales)
- El comportamiento del reproductor puede variar: Ante las mismas variaciones en un flujo de video, el reproductor puede variar su performance en lo que respecta a la calidad exhibida. Para mitigar este impacto es conveniente borrar el caché antes de cada corrida o incluso forzar la relectura de la configuración del *broadcaster* manager y reproducir más de una vez el mismo flujo
- Las pruebas de pérdidas por ráfagas deben ser consistentes con el tamaño de chunk y los cuadros por chunk, que pueden variar bastante entre diferentes fuentes de video. Es recomendable definir la opción de GoalBit *broadcaster* manager "-- *broadcaster-max-stucked-chunks* 0" para evitar que se bloquee ante ráfagas grandes.
- Sincronización y referencias de tiempo. La sincronización del peer con el servidor y las referencias de tiempo del flujo de video son fundamentales. Un error de referencia de tiempo en el flujo de video puede causar la reproducción fuera de tiempo, pérdida de parte del flujo o la no reproducción del flujo. Es recomendable probar con diferentes tipos de contenedores, ya que algunos son más robustos que otros en este sentido (para GoalBit ASF es en general el más conveniente).
- Concatenación de muestras de video. En este trabajo se optó por unir las muestras de video en una única tira de video, lo cual permite simplificar y agilizar la reproducción, minimizar la introducción de errores sistemáticos por parte de quien plantea la prueba (comparado con la presentación de 106 muestras individuales) y mantener la atención del observador lo más constante posible. La generación de esta tira implica introducir algunos cuadros de "relleno" en las pausas en negro para dar tiempo al reproductor a adaptarse ante un video con alta pérdida. También es necesario a veces completar al comienzo o al final de la muestra los cuadros faltantes para que sea compatible con la secuencia de GOPs, debido a que el corte manual del video original pudo perder algunos cuadros.

## V Análisis de resultados

Como se describió en el Capítulo anterior, en este trabajo se obtuvo una tira de videos (106 muestras) de 10s cada uno. Dichos videos presentan diversos trastornos de calidad, desde pixelado hasta congelamiento de imagen, todos introducidos por variaciones de las condiciones de red (ancho de banda, pérdida, tamaño medio de ráfaga de pérdidas y latencia) dentro de los rangos indicados para las redes ADSL y 3G en la caracterización brindada en III.5.

Para cada video de muestra se evalúan como parámetros de entrada candidatos (como se explicó en IV) las pérdidas y MLBS por tipo de cuadro (ver tabla con parámetros por muestra de video en Anexo 2). Graficando los resultados para el grupo de 106 videos de muestra se observa un cubrimiento razonable del rango de variación de pérdidas y ráfagas de pérdidas de I, P y B.



- Figura 5.1: Cubrimiento de rango de muestras de video seleccionadas –

Respecto a los rangos de variación, dada la escasa proporción de cuadros I en las secuencias de 10s, se consideró asumiendo probabilidades de pérdida de 0, 0.5 y 1, siendo lo razonable entre 0 y 0.5 debido al alto impacto en calidad. Para P se admitieron probabilidades de pérdida de hasta 0.3 y para B hasta 0.5.

Cabe señalar que dada la baja correlación observable entre los distintos parámetros evaluados, es posible considerarlos como entradas. Las pérdidas por tipo de cuadro son candidatos seguros, mientras que las ráfagas de pérdidas son a evaluar. A priori no parece un candidato aceptable MLBSI (ráfagas de pérdidas del cuadro I), ya que su variación es en un rango muy acotado, lo que no presentaría un gran aporte a la evaluación de calidad. Esto se puede inferir sencillamente porque en un GOP no se ven cuadros I sucesivos, por lo tanto no se observarán ráfagas en ese tipo de cuadro. El mlbsB es un posible candidato extra, pero el rango de variación observado en las muestras es estrecho. Esto se puede explicar nuevamente en el GOP, ya que B es un cuadro de pobre contenido gráfico, por lo tanto de baja frecuencia. El mlbsP parece el más aceptable como 4 parámetro de entrada, ya que presenta un rango más amplio de variación. Esto es también justificable por un GOP típico, donde P es el tipo de cuadro más frecuente en general, en donde se pueden observar más ráfagas. De todos modos se deberá validar con las pruebas de la red neuronal a ver si vale la pena el costo de introducir este nuevo parámetro en la calidad del ajuste de la función de QoE.

### V.1 Formulando la calidad de experiencia

Se consideraron 10 evaluadores para las 106 muestras de video, obteniendo luego una “evaluación media”, la cual fue considerada como salida de la función de calidad PSQA a obtener (ver las evaluaciones en la tabla del Anexo 2).

Se entrenó una red neuronal con las entradas normalizadas e invertidas, para respetar las características normales de estas funciones. Considerando las observaciones realizadas en la sección anterior sobre las entradas candidatas y probando los resultados de error cuadrático medio, se seleccionaron 1-Lri, 1-Lrp, 1-Lrb y 1-10/mlbsP como entradas.

En forma comparativa se modeló la función PSQA en 2 y 3 capas. La función en 2 capas, sin neuronas intermedias, sirve por permitir una expresión polinomial de la calidad de servicio, mientras que la función de 3 capas, mejora la estimación de calidad, a costa de agregar un poco de complejidad de implementación.

Recogiendo los resultados de [1], la función de calidad para una RNN de 2 niveles, de parámetros  $\lambda_i$  es:

$$Q(\lambda_i) = \frac{\sum \lambda_i \cdot a(\lambda_i)}{[\sum (b(\lambda_i) \cdot \lambda_i) + v]}$$

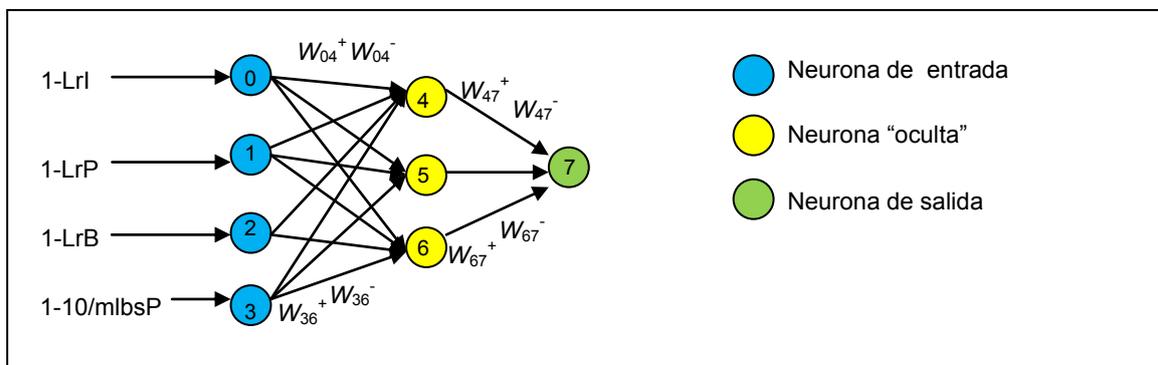
donde v se fija en 0.01 y los coeficientes son los de la tabla 5.1:

a(Lri)	0,508517	b(Lri)	0,491483
a(Lrp)	0,929675	b(Lrp)	0,070325
a(Lrb)	0,000000	b(Lrb)	1,000000
a(mlbsP)	0,000000	b(mlbsP)	1,000000

- Tabla 5.1: Coeficientes de la fórmula de calidad para una RNN de 2 niveles –

El error cuadrático medio obtenido usando esta función es de 0.0195.

Para la función que usa una red neuronal de 3 capas, la función de calidad viene dada por las matrices de pesos relativos de transición entre las neuronas. La figura 5.2 muestra un diagrama que facilita la identificación de dichos pesos y luego las tablas 5.2a y b indican los valores para cada uno. El error cuadrático medio obtenido en este caso es de 0.0187.



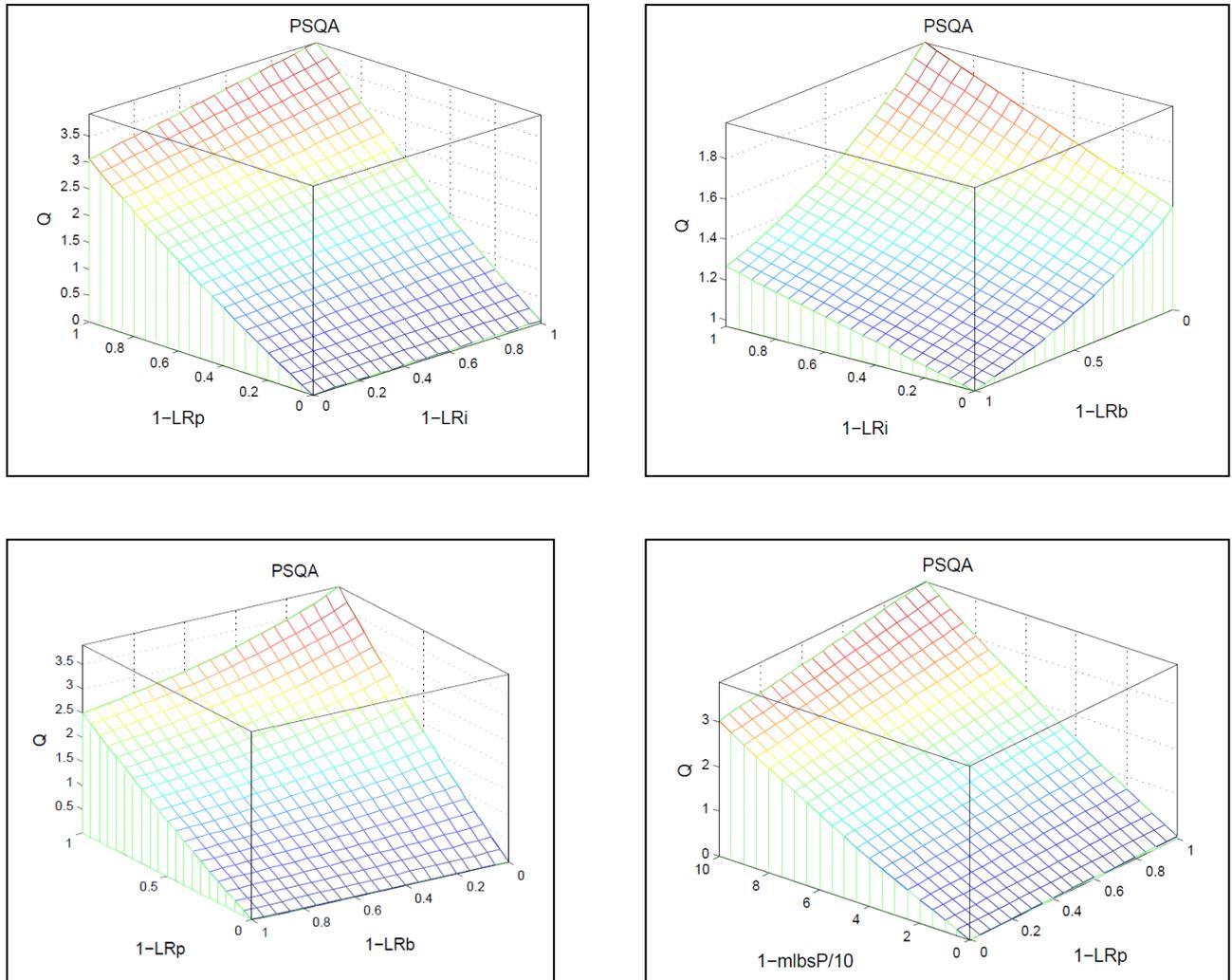
- Figura 5.2: Diagrama de RNN 3 capas implementada –

Wij+	0	1	2	3	4	5	6	7
0	0	0	0	0	0.0119669	0.0357098	0.0119239	0
1	0	0	0	0	0.0185474	285.307	0.0184762	0
2	0	0	0	0	279.084	0	211.915	0
3	0	0	0	0	343.391	0	29.284	0
4	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	509.409
6	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0

Wij-	0	1	2	3	4	5	6	7
0	0	0	0	0	47732	0.0038064	359639	0
1	0	0	0	0	365226	0.0116028	362733	0
2	0	0	0	0	260.853	0	281.265	0
3	0	0	0	0	203.867	0	206.195	0
4	0	0	0	0	0	0	0	12657
5	0	0	0	0	0	0	0	0.0285381
6	0	0	0	0	0	0	0	114049
7	0	0	0	0	0	0	0	0

- Tablas 5.2a y b: Pesos relativos de transición (positiva y negativa) entre neuronas –

Los resultados obtenidos para esta función de tres capas se pueden observar a partir de las gráficas de respuesta de la figura 5.3.



- Figura 5.3: Respuesta de la función PSQA para red neuronal de 3 capas -

El error cuadrático medio obtenido en este caso es de 0.0187. Observar que la respuesta es cualitativamente aceptable, puesto que cumple con la monotonía y la sensibilidad ante las diferentes entradas. Se puede observar que las pérdidas de cuadros P son las de mayor impacto y esto es razonable debido a la gran proporción de éstos en el flujo de video. Pero es a su vez notable el impacto de las pérdidas en ráfagas de cuadros P. Corroborando lo que es razonable a priori: que tiene un impacto mucho mayor la ráfaga que la pérdida aislada.

La baja sensibilidad en Lrl dada la importancia de este tipo de cuadro puede resultar extraña, aunque se debe analizar considerando al menos dos factores. Primero, que la comparación es entre perder un cuadro I y perder varios cuadros P o B, puesto que P o B se pierden generalmente en ráfagas, o al menos varios cuadros del mismo tipo cuando se evalúan pérdidas de red (un chunk perdido contiene varios P y B siempre,

aunque puede no contener cuadros I). Y segundo, que la probabilidad de pérdida en I (dada la baja frecuencia de estos cuadros en una muestra de 10s) fue ajustada a tomar valores específicos 0, 0.5 y 1, lo cual tiene un error implícito. Claro que un muestreo con mayor cantidad de videos con pérdidas en este tipo de cuadro permitiría una mejor aproximación de la función de calidad.

Finalmente se observará que los rangos de variación de la respuesta no alcanzan el máximo. Esto se debe a que las muestras consideradas se focalizan en la evaluación de la calidad media y mala del video, donde tiene un mejor ajuste y no se forzó la respuesta natural de calidad 5 porque empeoraba el error cuadrático medio. Es esperable sin embargo que un muestreo más amplio de videos incluyendo más de aquellos con alta calidad mejore la respuesta.

- Página en blanco, intencional -

## VI Conclusiones

En este trabajo se obtuvo una formulación de la calidad de experiencia de video ante una red de transporte de calidad variable (con latencia, jitter, pérdidas y ráfagas de pérdidas), que es aceptado como unos de los principales factores de incidencia en la QoE de aplicaciones de video.

Se planteó una plataforma de pruebas sencilla y flexible que permitió simular mediante NetEm2 las condiciones que pueden ser observadas en redes populares como ADSL o 3G, planteando variaciones en latencia, jitter, niveles de pérdida y ráfagas de pérdida.

Se analizaron diversos factores de los flujos de video MPEG-4 parte10/H.264, pasando por elementos básicos como los VOP (cuadros I, P y B), pero evaluando también la tasa de transferencia, el tamaño de imagen, el ancho de banda de codificación de video y audio, o la selección el multiplexor más adecuado para las redes mencionadas y la aplicación de video GoalBit.

Se estudiaron las características más relevantes de GoalBit para la transmisión de video en esas redes, como ser la tasa de transmisión, la transcodificación, los multiplexores, el tamaño de chunk o el control de ventana de chunks. Haciendo uso de estos parámetros y funcionalidades incluidas en Goalbit se generó un conjunto de herramientas auxiliares y complementarias (en forma de scripts) para simplificar y automatizar lo más posible la obtención de las secuencias de muestra y datos de prueba.

Recurriendo a la metodología PSQA y Redes Neuronales, se obtuvieron dos formulaciones de la calidad de experiencia: por un lado una función explícita de calidad como cociente de polinomios de modo de facilitar la implementación, y por otro una red neuronal de tres niveles que si bien es más compleja, mejora la precisión. Esta función es dependiente de las pérdidas de cuadros I,P,B y de las ráfagas de pérdidas en P únicamente, considerando que el impacto de las ráfagas en las otros cuadros es poco significativo y que una función de 4 entradas es razonablemente compleja.

Este trabajo, dado que considera como parámetros de entrada las pérdidas y ráfagas de pérdidas por tipo de cuadro, puede ser utilizado como base para la evaluación de QoE en plataformas P2P como GoalBit. Si bien está enfocado en pérdidas de red, los pérdidas por cuadro y su rango de variación puede aplicarse también a otras causas de pérdida como fallas en un peer "fuente" o incluso desconexión temporal del mismo. Incluso se puede incrementar la complejidad el modelo de red de transporte plateando pérdidas en ráfagas con densidad variable, algo que NetEm2 permite pero no fue probado explícitamente en este trabajo.

Finalmente es claro que la precisión de los resultados se puede mejorar aumentando la cantidad de muestras de video y la cantidad de observadores. En este caso por restricciones de tiempo del estudio para la cantidad de muestras se consideró un cubrimiento mínimo en el rango de interés de variación de los parámetros de entrada

como se planteó en el Capítulo V. No obstante si se desearan agregar parámetros de red más complejos, la cantidad de muestras debe aumentar. Por el mismo motivo la cantidad de observadores se fijó en 10, observando a la vez que la desviación de las valoraciones era aceptable.

## Referencias

- [1] P. Rodríguez-Bocca. Quality-centric design of *Peer-to-Peer* systems for live-video broadcasting. Tesis PhD Universidad de la República - Facultad de Ingeniería, Febrero 2008.
- [2] ITU-R Recommendation BT.500-11. Methodology for the subjective assessment of the quality of television pictures, ITU recommendation, Junio 2002.
- [3] C.J. van den Branden Lambrecht. Perceptual Models and Architectures for Video Coding Applications. Tesis PhD, EPFL, Lausanne, 1996.
- [4] P. Rodríguez-Bocca, G. Rubino, F. Robledo, D. de Vera. GoalBit: The First Free and Open Source *Peer-to-Peer* Streaming Network, Mayo 2009.
- [5] J. Fabini, P. Reichl, A. Poropatich. Measurement-Based Modeling of NGN Access Networks from an Application Perspective, Abril 2008
- [6] S. Hemminger. Network Emulation with NetEm, Abril 2005.
- [7] G. Almes, S. Kalidindi, M. Zekauskas. RFC 2679: A One-way Delay Metric for IPPM, Setiembre 1999.
- [8] G. Almes, S. Kalidindi, M. Zekauskas. RFC 2680: A One-way Loss Metric for IPPM, Setiembre 1999.
- [9] The Wireless Communications and Pervasive Computing Team. Video Streaming performance over GPRS/UMTS networks, Nov 2008.
- [10] Kikuchi, Nomura, Fukunaga. RFC 3016: RTP Payload Format for MPEG-4 Audio/Visual Streams, Noviembre 2000.
- [11] J. Ehrensberger, A. Revuelta, J-R. Schuler. Video on Demand over the Internet, Noviembre 2006
- [12] J. A. Berná Galiano, M. F. Pérez Polo. Estudio de retardos en Internet para el control de sistemas de red, 2006
- [13] S. Salsano, F. Ludovici, A. Ordine. Definition of a general and intuitive loss model for packet networks and its implementation in the NetEm module in the Linux kernel, Julio 2009.
- [14] S. Gnani, M. Grangetto, E. Magli, G. Olmo. Rate allocation for video transmission over lossy correlated networks, Electronics Letters, Setiembre 2002.
- [15] Sitio web de NetEm2: <http://netgroup.uniroma2.it/twiki/bin/view.cgi/Main/NetEm2>, 2009.
- [16] Foro Ubuntu Master Kernel: <http://ubuntuforums.org/showthread.php?t=311158>, 2009.
- [17] Proyecto Kernel Check: <http://sourceforge.net/projects/kcheck>, 2009.

[18] Wiki Proyecto GoalBit:

<http://sourceforge.net/apps/mediawiki/goalbit/index.php?title=DebianCompile>, 2009.

[19] Qi Qu, Yong Pei, J. W. Modestino, X. Tian. Source-Adaptation-Based Wireless Video Transport: A Cross-Layer Approach, Agosto 2005.

[20] Estándares ADSL: ANSI T1.413 Issue 2, ITU G.992.1, ITU G.992.2 e ITU G.994.1, Noviembre 2007.

[21] Estándares ADSL2/2+: ITU G.992.3 e ITU G.992.5, Enero 2009

[22] Especificaciones generales del 3GPP sobre UMTS FDD. Estándar TS 25.101, <http://www.3gpp.org/article/umts>, Diciembre 2006

[23] Sitio oficial UMTS World: <http://www.umtsworld.com/>, Setiembre 2006

[24] S. Mohamed, G. Rubino. A Study of Real-time Packet Video Quality Using Random Neural Networks. IEEE Transactions On Circuits and Systems for Video Technology, 12(12):1071 –1083, Diciembre 2002.

[25] G. Rubino, M. Varela, and J-M. Bonnin. Controlling multimedia QoS in the future home network using the PSQA metric. submitted to IEEE Trans. on Systems Man and Cybernetics part B, 2005.

[26] E. Gelenbe. Random Neural Networks with Negative and Positive Signals and Product Form Solution. Neural Computation, 1(4):502–511, 1989.

[27] Estándar MPEG-1 o MP3: ISO/IEC 11172-3, 1993

[28] Estándar MPEG-2, ISO/IEC 13818. Information technology generic coding of moving pictures and associated audio information, Enero 2005.

[29] D. Marshall. "MPEG-2, MPEG-3, and MPEG-4", Abril 2001.

[30] Estándar H.264 o MPEG4-10: ITU-T H.264 o ISO/IEC MPEG-4 AVC (ISO/IEC 14496-10 - MPEG-4 Part 10 Advanced Video Coding), Octubre 2009.

[31] Estándar MPEG-7: Information technology -- Multimedia content description interface -- Part 8: Extraction and use of MPEG-7 descriptions, ISO/IEC TR 15938-8:2002, Diciembre 2002.

[32] Estándar MPEG-21: Information technology -- Multimedia framework (MPEG-21) -- Part 3: Digital Item Identification , ISO/IEC 21000-3:2003 , Marzo 2003.

[33] Estándar ITU-T H.261. Video codec for audiovisual services at px64 kbit/s, Enero 1993.

[34] Estándar ITU H.262. Information technology - Generic coding of moving pictures and associated audio information: Video. Febrero 2000.

[35] ITU-T Standard H.263. Video coding for low bit rate communication, Enero 1998.

[36] Mean Opinion Score (MOS), recomendación ITU-T P.800. Methods for Subjective Determination of Transmission Quality, 1996.

- [37] ITU-T Recommendation P.910. Subjective video quality assessment methods for multimedia applications, 2000.
- [38] Recomendación ITU-T P.920. Interactive test methods for audiovisual communications, 2000.
- [39] P.N. Tudor. MPEG-2 Video Compression. Electronics & Engineering Journal, Diciembre 1995.
- [40] S. A. Basith, S. R. Done. Digital Video, MPEG and Associated Artifacts, Junio 1996.
- [41] Estándar de grabación DVD. ISO 9660: 1988 Volume and file structure of CD-ROM for information interchange, Abril 1988.
- [42] Sitio oficial de promoción de formato DVD: <http://www.dvdforum.org/forum.shtml>, Enero 2010.
- [43] Benchmarking Speed test: <http://www.speedtest.net/>, Enero 2010
- [44] Broadband reviews: <http://www.broadband.org/>, Enero 2010
- [45] R. Braden. Requirements for Internet Hosts -- Communication Layers: RFC 1122, Octubre 1989.
- [46] A. S. Tanembaun. Computer Networks (4a edición), Prentice Hall PTR.
- [47] Jiří Kulhánek. Image Quality Evaluation by Human Visual System Model. In 8th International Student Conference on Electrical Engineering, Mayo 2004.
- [48] HR Sheikh - AC Bovik. IEEE Transactions on Image Processing, 2006
- [49] Q. Huynh-Thu, M. Ghanbari. Scope of validity of PSNR in image/video quality assessment. Electronics Letters 44: 800–801, 2008
- [50] S. Cui, A. J. Goldsmith, A. Bahai. Energy-efficiency of MIMO and Cooperative MIMO in Sensor Networks. IEEE J. Select. Areas of Commun. 22 (6): 1089–1098, Agosto 2004
- [51] L. Zheng , D. N. C. Tse. Diversity and multiplexing: A fundamental tradeoff in multiple-antenna channels". IEEE Trans. Inf. Th. 49 (5): 1073–1096. Mayo 2003.
- [52] Zi Ling, Ivan Lee. Adaptive Multi-Path Video Streaming, ism, pp.399-406, Eighth IEEE International Symposium on Multimedia (ISM'06), Diciembre 2006.
- [53] I. Lee, L. Guan. Reliable video communication with multi-path streaming using MDC, IEEE Multimedia and Expo, Julio 2005.
- [54] A. MacAulay, B. Felts, Y. Fisher. IP Streaming of MPEG-4: Native RTP vs MPEG-2 Transport Stream, Envivio whitepaper. Octubre 2005.
- [55] Sitio web Envivio: [www.envivio.com/](http://www.envivio.com/), Enero 2010.

- [56] Sitio web Alcatel-Lucent: [www.alcatel-lucent.com/](http://www.alcatel-lucent.com/), Enero 2010.
- [57] Sitio web Ericsson: [www.ericsson.com/](http://www.ericsson.com/), Enero 2010.
- [58] M. Steiner, E. W. Biersack, T. En-Najjary. Exploiting KAD: Possible Uses and Misuses. *Computer Communication Review*, 37(5), Octubre 2007.
- [59] A. Binzenhöfer, H. Schnabel. Improving the Performance and Robustness of Kademlia-based Overlay Networks, KIVS 2007, Febrero 2007.
- [60] T. Cholez, I. Chrisment, O. Festor. Evaluation of Sybil Attacks Protection Schemes in KAD, Scalability of Networks and Services, Junio 2009.
- [61] S. Mohamed. Automatic Evaluation of Real-Time Multimedia Quality: a Neural Network Approach. Tesis PhD, INRIA/IRISA, Univ. Rennes I, Rennes, Francia, Enero 2003.
- [62] S. Mohamed, F. Cervantes, and H. Afifi. Audio Quality Assessment in Packet Networks: an Inter-Subjective Neural Network Model. In Proc. of IEEE International Conference on Information Networking (ICOIN-15), pag. 579 –586, Beppu, Oita, Japón, Enero 2001.
- [63] P. Romero. Selección de piezas eficiente para un sistemas P2PTV. Maestría en Ingeniería (Ingeniería Matemática), Universidad de la República - Facultad de Ingeniería. 2009.
- [64] D. Padula. Selección de pares eficiente geográficamente para un sistemas P2PTV. Maestría en Ingeniería (Ingeniería Matemática), Universidad de la República - Facultad de Ingeniería. En ejecución.
- [65] L. Stábile. Distribución de video en vivo desde múltiples fuentes. Tesis/Monografía de grado, Ingeniería en Computación, Universidad de la República - Facultad de Ingeniería. 2006.
- [66] J. L. Fernández, D. De Vera, A. Chadarevian. Redes de Distribución de Video: monitoreo y estadísticas. Tesis/Monografía de grado, Ingeniería en Computación, Universidad de la República - Facultad de Ingeniería. 2006.
- [67] P. Rodriguez-Bocca, D. De Vera, G. Rubino. QoE Monitoring Platform for Video Delivery Networks, IP Operations and Management, Noviembre 2007.
- [68] D. De Vera. Perceived Quality in Mesh-based P2PTV Systems. Maestría en Informática, Universidad de la República - Facultad de Ingeniería. 2008.
- [69] Sitio web de NIST Net: <http://snad.ncsl.nist.gov/nistnet/>, 2005.
- [70] Sitio web de Dummynet: <http://info.iet.unipi.it/~luigi/dummynet/>, 2009
- [71] Sitio web de Network Simulator (NS) versión 3 : <http://www.nsnam.org/index.html>, Enero 2010.
- [72] M. Varela Rico. Pseudo-subjective Quality Assessment of Multimedia Streams and its Applications in Control. Tesis PhD INRIA/IRISA, Univ. Rennes I, Rennes, Francia, Noviembre 2005.

- [73] M. Bernaschi, F. Cacace, R. Clementelli, L. Vollero. Adaptive Streaming On Heterogeneous Networks. 2005
- [74] T. Nguyen, A. Zakhor. Distributed Video Streaming Over Internet. Proceedings of SPIE/ACM Multimedia Computing and Networking (MMCN2002), Enero 2002.
- [75] J.A. Pouwelse, P. Garbacki, D.H.J. Epema, H.J. Sips. The BitTorrent P2P file sharing system: measurements and analysis. *Peer-to-Peer Systems IV*. Noviembre 2005.
- [76] TeliaSonera International Carrier. Content delivery networks: Ensuring quality of experience in streaming media applications, Mayo 2009.
- [77] K. Piamrat, C. Viho, J-M Bonnin, A. Ksentini. Quality of Experience Measurements for Video Streaming over Wireless Networks. Sixth International Conference on Information Technology: New Generations, Abril 2009.
- [78] H. Van Peteghem, L. Schumacher. Implementation of an Open-Source UTRAN Testbed. Noviembre 2006.
- [79] 3GPP. HSPA: <http://www.3gpp.org/HSPA>, 2009.
- [80] 3GPP. Especificación TR 25.999: High Speed Packet Access (HSPA) evolution; Frequency Division Duplex (FDD). Setiembre 2007.
- [81] J. Peisa, S. Wager, M. Sågfors, J. Torsner, B. Göransson, T. Fulghum, C. Cozzo y S. Grant. High Speed Packet Access Evolution – Concept and Technologies, Ericsson, 2007.
- [82] Karl J. Molnar y Stephen J. Grant. Performance of Dedicated Indoor MIMO HSDPA Systems, Ericsson, 2008.
- [83] Ericsson. The Evolution of EDGE, Febrero 2007
- [84] F. Hartung, U. Horn, J. Huschke, M. Kampmann, T. Lohmar y M. Lundevall. Delivery of Broadcast Services in 3G Networks, IEEE Transactions on Broadcasting, vol. 53, No. 1, Marzo 2007.
- [85] Nortel Networks: <http://www.nortel.com/>. Mobile Backhaul Evolves with Carrier Ethernet, Setiembre 2008.
- [86] S. Mohamed, G. Rubino y M. Varela. Performance evaluation of real-time speech through a packet network: a Random Neural Networks-based approach, Mayo 2004.
- [87] Sitio Antel Uruguay: [www.antel.com.uy/](http://www.antel.com.uy/), Enero 2009.
- [88] Ericsson. IP Transport Solution for CS Traffic, 2006.
- [89] Dah M. Chiu, R. W. Yeung, Jiaqing Huang y Bin Fan. Can network coding help in p2p networks? In Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks, 2006 4th International Symposium on, pag. 1–5, 2006.
- [90] Xi Li, Y. Zeng, B. Kracker, R. Schelb, C. Goerg y A. Timm-Giel. Carrier Ethernet for Transport in UMTS Radio Access Network: Ethernet Backhaul Evolution, Noviembre 2006.

- [91] A. P. Couto da Silva, P. Rodríguez-Bocca y G. Rubino. Coupling QoE with dependability through models with failures. In 8th International Workshop on Performability Modeling of Computer and Communication Systems (PMCCS'07), Edinburgo, Escocia, Setiembre 2007.
- [92] A. P. Couto da Silva, P. Rodríguez-Bocca y G. Rubino. Optimal quality-of-experience design for a p2p multi-source video streaming. In IEEE International Conference on Communications (ICC 2008), Beijing, China, Mayo 2008.
- [93] X. Hei, C. Liang, J. Liang, Y. Liu y K.W. Ross. Insights into p2p live: A measurement study of a large-scale p2p iptv system. In Proc. of IPTV Workshop, conferencia web internacional, 2006.
- [94] Q. Huang, H. Jin y X. Liao. P2P live streaming with tree-mesh based hybrid overlay. In ICPPW '07: Proceedings of the 2007 International Conference on Parallel Processing Workshops (ICPPW 2007), page 55, Washington, DC, USA, 2007.
- [95] P. Simoens et al. Towards Autonomic Access Networks for Service QoE Optimization. 1st IEEE International Workshop on MACE, Octubre 2006.
- [96] M. Martínez, A. Morón, F. Robledo, P. Rodríguez-Bocca, H. Cancela y G. Rubino. A GRASP Algorithm Using RNN for Solving Dynamics in a P2P Live Video Streaming Network, Proceedings of the 2008 8th International Conference on Hybrid Intelligent Systems, pag 447-452, 2008.
- [97] M. Ghareeb y C. Viho. Performance Evaluations of a QoE-Based Multipath Video Streaming Mechanism over Video Distribution Network (VDN), Future Multimedia Networking, Junio 2009.
- [98] P. Reichl. From 'Quality-of-Service' and 'Quality-of-Design' to 'Quality-of-Experience': A Holistic View on Future Interactive Telecommunication Services, 2007.
- [99] K. Deep Singh. Improving Quality of Service and Resource Utilization for Multimedia Streaming over Third Generation Mobile Networks, Tesis PhD Universidad de Rennes I, Diciembre 2007.
- [100] M. Davies, A. Dantcheva y P. Fröhlich. Comparing access methods and quality of 3g mobile video streaming services. Conference on Human Factors in Computing Systems, 2008.
- [101] L. Superiori et al. Content-Aware Scheduling for Video Streaming over HSDPA Networks. 2009.
- [102] J. Greengrass, J. Evans y A.C. Begen. Not All Packets Are Equal, Part I: Streaming Video Coding and SLA Requirements. Internet Computing, IEEE, Enero 2009.
- [103] Y. Deng et al. A novel network-adaptive buffering mechanism for video streaming based on feedback. China-Ireland International Conference on Information and Communications Technologies (CIICT 2008), p.775-780, Setiembre 2008.
- [104] E. N. Gilbert, "Capacity of a Burst-Noise Channel," Bell Systems Technical Journal, vol. 39, pp. 1253-1265, Setiembre 1960.

- [105] E. O. Elliot, "Estimates of Error Rates for Codes on Burst-Noise Channels," Bell Systems Technical Journal, vol. 42, pp. 1977–1997, Setiembre 1963.
- [106] Especificaciones Kademila:  
<http://xlattice.sourceforge.net/components/protocol/kademlia/specs.html>, 2008.
- [107] B. Cohen. Incentives build robustness in bittorrent. Workshop Economics of Peer-to-Peer Systems, Berkeley, EEUU, Mayo 2003. <http://bittorrent.com/>.
- [108] Sitio oficial de Matlab (MATrix LABoratory): <http://www.mathworks.com/>, 2010.
- [109] Sitio oficial de Avidemux: <http://fixounet.free.fr/avidemux/>, 2010.
- [110] Sitio oficial de x264 (librería abierta de h.264/MPEG-4 desarrollada por VideoLan): <http://www.videolan.org/developers/x264.html>, 2010.
- [111] I. Politis, M. Tsagkaropoulos, T. Dagiuklas y S. Kotsopoulos. Study of the QoS of Video Traffic over Integrated 3GWLAN systems, 2006.
- [112] D. Kerr. Chrominance Subsampling in Digital Images, Diciembre 2009.
- [113] Sitio oficial de VLC (reproductor multimedia desarrollado por VideoLan): <http://www.videolan.org/vlc/>, 2010.
- [114] Sitio oficial de DirectTV (Cadena de distribución de TV vía satélite): <http://www.directv.com>, 2010.

## Glosario

ABI	Active Buffer Index
ADSL	Asymmetric Digital Subscriber Line
DL	Downlink
EDGE	Enhanced Data Rates for GSM Evolution
GBPS	GoalBit Packetized Stream
GBTP	Goalbit Transport Protocol
GOP	Group Of Pictures
GPRS	General Packet Radio Service
GSM	Global System for Mobile communications
HDTV	High Definition TeleVision
HSDPA	High Speed Downlink Packet Access
HSUPA	High Speed Uplink Packet Access
HVS	Human Vision System
ISO	International Organization for Standardization
ITU	International Telecommunication Union
LR	Loss Rate
MDC	Multi Description Coding
MIMO	Multiple-Input Multiple-Output
MLBS	Mean Loss Burst Size
MPEG	Moving Picture Experts Group
MTU	Maximum Transport Unit
OSI	Open System Interconnection
P2P	Peer to Peer
PSNR	Peak Signal to Noise Ratio
PSQA	Pseudo Subjective Quality Assessment
QoE	Quality of Experience
RAB	Radio Access Bearers
RNN	Random Neural Networks
UL	Uplink
UMTS	Universal Mobile Telecommunications System
WLAN	Wireless Local Area Network

## Anexo 1 Configuración de nodo con NetEm2

Para configurar una máquina con NetEm2 se deben cumplir los siguientes requisitos. En este anexo se plantea desde cero, con Ubuntu 8.04 que era la distribución más fácilmente disponible.

### Requisitos de NetEm2

- 1) Sistema operativo Linux con packet forwarding
- 2) Actualizar el Kernel a versiones soportadas según autor [15]: 2.6.29, 2.6.30 o 2.6.31
- 3) Instalar iproute2
- 4) Instalar el parche de NetEm2

### Paso 1

Para habilitar packet forwarding primero hay que definir dos interfaces de red. Esto se hace o bien gráficamente con el editor de red de Ubuntu o en el archivo `/etc/network/interfaces`:

```
auto lo
iface lo inet loopback

auto eth0 eth1
iface eth0 inet static
    address 192.168.151.135
    netmask 255.255.255.0
    broadcast 192.168.151.255
    network 192.168.151.0

iface eth1 inet static
    address 192.168.2.1
    netmask 255.255.255.0
    broadcast 192.168.2.255
    network 192.168.2.0
```

Luego para habilitar port forwarding se procede de la siguiente manera:

```
# echo 1 > /proc/sys/net/ipv4/ip_forward

O bien editar en la configuración del Kernel ip_forward con el valor 1:

# vi /etc/sysctl.conf

net.ipv4.ip_forward = 1
```

## Paso 2

Para actualizar el kernel existen 2 opciones: utilizar la aplicación Kernelcheck o hacer la instalación manual siguiendo los buenos pasos descritos en Ubuntu forums [17]. Ambas opciones son válidas y seguras. En este trabajo se realizó una actualización manual para intercalar la actualización de NetEm2 antes de compilar.

Obtener las Fuentes del Kernel y las herramientas de compilación.

```
$ cd /usr/src
$ sudo apt-get install build-essential bin86 kernel-package libqt3-headers
libqt3-mt-dev wget libncurses5 libncurses5-dev
```

Pasar a usuario root y obtener la imagen del Kernel

```
$ sudo -s
# wget -c http://kernel.org/pub/linux/kernel/v2.6/linux-2.6.31.tar.bz2 &&
tar -xvjf linux-2.6.31.tar.bz2
```

Modificar el link del Kernel hacia el nuevo directorio de Linux y moverse al directorio

```
# rm -rf linux && ln -s /usr/src/linux-2.6.31 linux && cd /usr/src/linux
```

Aplicar algunos parches recomendados:

```
# wget -c http://kernel.org/pub/linux/kernel/v2.6/patch-2.6.31.4.bz2
# bzcat patch-2.6.31.4.bz2 | patch -p1
```

Aplicar el parche de NetEm2

```
# patch -p1 < ~/Desktop/netem_patch/netem2_patch
```

Saltar al Paso 3 y luego continuar con la configuración del Kernel. De esta forma agregamos todos los parches al mismo tiempo, minimizando el tiempo de instalación.

Configurar el Kernel: hay muchas opciones personalizables, se recomienda prestar atención a la placa de red inalámbrica (devices -> wireless). Configurar NetEm2 como un módulo (marcar con M).

```
# cp /boot/config-$(uname -r) .config && yes "" | make oldconfig
# make menuconfig
```

Preparar ejecutable del Kernel para compilar (parado en /usr/src/linux)

```
# make-kpkg clean

# INSTALL_MOD_STRIP=1 CONCURRENCY_LEVEL=3 make-kpkg --initrd \
--append-to-version=-mk kernel_image \
kernel_headers modules_image
```

E instalar (en /usr/src)

```
# cd .. && dpkg -i linux*2.6.31*.deb
```

Paso 3

Para instalar iproute 2 es necesario bajar las fuentes (al igual que para la actualización del kernel) y las build-dep.

```
# cd /usr/src
# git clone
git://git.kernel.org/pub/scm/linux/kernel/git/shemminger/iproute2.git
# apt-get install build-dep iproute
```

Instalar el parche de NetEm2 (asumiendo que está en ~/Desktop/netem\_patch)

```
# cd /usr/src/iproute2
# patch -p1 < ~/Desktop/netem_patch/tc_patch
```

Luego compilarlo e instalarlo

```
# make
# make install
```

### Ejecución de NetEm2

NetEm2 se debe levantar como un módulo:

```
# modprobe sch_netem2
```

Para desactivar el módulo:

```
# modprobe -r sch_netem2
```

El comando sigue siendo *tc*, sólo que si quedaron NetEm y NetEm2, se debe indicar con cuál se trabaja, por lo tanto usar */usr/sbin/tc* para NetEm2. Un ejemplo para aplicar un retardo de 10s con 2s de jitter, es el siguiente:

```
# /usr/sbin/tc qdisc add dev eth0 root netem2 delay 10 2
```

## Anexo 2 Datos de las muestras de video

### Parámetros de las muestras de video

Video	indice_sec	Perdida			mlbsI	mlbsP	mlbsB	Orden
		Lri	Lrp	Lrb				
ski	2601	0,000	0,000	0,000	0,000	0,000	0,000	0
ski	145	0,000	0,154	0,112	0,000	1,000	1,000	1
ski	456	0,500	0,168	0,184	1,000	1,000	1,000	2
ski	636	0,000	0,138	0,188	0,000	1,000	1,000	3
ski	828	0,000	0,083	0,151	0,000	1,000	1,000	4
ski	987	0,000	0,104	0,123	0,000	1,000	1,000	5
ski	1153	0,500	0,100	0,260	1,000	1,000	1,000	6
ski	1355	0,000	0,114	0,159	0,000	1,000	1,000	7
ski	1971	0,000	0,147	0,416	0,000	1,000	1,000	8
ghost	1439	0,000	0,048	0,048	0,150	1,170	1,000	9
spider	2158	0,000	0,050	0,150	0,000	1,000	1,000	10
ghost	4100	0,000	0,009	0,047	0,620	1,000	1,000	11
ghost	1939	0,000	0,117	0,117	0,250	1,450	1,000	12
ghost	3864	0,000	0,050	0,130	0,000	1,000	1,000	13
spider	675	0,000	0,076	0,210	0,000	1,000	1,000	14
spider	1125	0,000	0,160	0,300	1,000	1,000	1,000	15
spider	1358	0,000	0,200	0,220	0,000	1,000	1,000	16
spider	1716	0,500	0,110	0,130	1,000	1,000	1,000	17
spider	1985	0,000	0,060	0,070	0,000	1,000	1,000	18
spider	2288	0,500	0,200	0,220	1,000	1,000	1,000	19
ghost	3739	0,000	0,084	0,075	0,340	1,380	1,000	20
path		0,000	0,000	0,000	0,000	0,000	0,000	21
scary	859	0,000	0,020	0,020	0,000	1,000	0,880	22
scary	1586	0,000	0,019	0,025	0,000	1,000	1,000	23
scary	1810	0,000	0,010	0,042	0,000	1,000	1,000	24
scary	2406	0,000	0,020	0,190	0,000	1,000	0,830	25
scary	3004	0,000	0,017	0,027	0,000	1,000	1,000	26
scary	3579	0,000	0,029	0,025	0,000	1,080	0,950	27
scary	3884	0,000	0,064	0,047	0,000	1,330	1,000	28
scary	1530	0,500	0,022	0,016	1,000	1,000	1,000	29
scary	3649	0,000	0,066	0,051	0,000	1,630	1,250	30
scary	1025	0,000	0,111	0,040	0,000	1,930	1,000	31
ciudad	1687	0,000	0,090	0,078	0,000	1,000	1,300	32
ciudad	2052	0,000	0,107	0,088	0,000	1,360	1,400	33
ciudad	3356	0,500	0,130	0,090	1,000	1,380	1,500	34
ciudad	2971	0,000	0,074	0,052	0,000	1,140	2,000	35
ciudad	3725	0,500	0,012	0,023	1,000	1,000	1,000	36
ski	3278	0,000	0,107	0,159	0,000	1,920	1,430	37
ski	1841	0,000	0,014	0,037	0,000	1,000	1,000	38
ski	886	0,000	0,036	0,087	0,000	1,000	1,000	39
ski	3158	0,000	0,054	0,146	0,000	1,440	1,600	40
ski	1153	0,000	0,166	0,139	0,000	1,000	1,000	41
futbol		0,000	0,000	0,000	0,000	0,000	0,000	42
path	1323	0,000	0,038	0,066	0,000	1,040	1,110	43
path	1570	0,000	0,060	0,140	0,000	1,200	1,850	44
path	806	0,000	0,025	0,011	0,000	1,000	1,000	45

path	3911	0,000	0,025	0,028	0,000	1,018	1,060	46
path	4290	0,000	0,077	0,108	0,000	1,200	1,670	47
path	2981	0,500	0,021	0,000	1,000	1,000	0,000	48
path	2636	0,000	0,044	0,061	0,000	1,450	1,130	49
path	1398	0,500	0,040	0,073	1,000	1,800	1,000	50
path	1135	0,500	0,077	0,065	1,000	1,440	1,143	51
path	3351	0,000	0,044	0,031	0,000	1,670	1,000	52
spider	426	0,000	0,067	0,055	0,000	1,000	1,000	53
spider	875	0,000	0,091	0,059	0,000	1,000	1,000	54
spider	1288	0,500	0,086	0,058	1,000	1,000	1,000	55
spider	2025	0,000	0,072	0,161	0,000	1,000	1,000	56
spider	1751	0,500	0,050	0,090	1,000	1,000	1,000	57
futbol	25371	0,000	0,021	0,017	0,000	1,000	0,892	58
futbol	6515	0,000	0,676	0,555	0,000	3,162	1,944	59
futbol	15673	0,000	0,017	0,025	0,000	1,000	1,000	60
futbol	33357	0,000	0,105	0,037	0,000	1,000	1,000	61
futbol	1197	0,000	0,031	0,000	0,000	1,000	0,000	62
ghost		0,000	0,000	0,000	0,000	0,000	0,000	63
futbol	5854	0,000	0,020	0,020	0,000	1,000	1,000	64
futbol	3732	0,500	0,020	0,010	1,000	1,000	1,000	65
futbol	8629	0,000	0,039	0,033	0,000	1,110	0,934	66
futbol	17745	0,500	0,022	0,088	1,000	1,000	1,000	67
futbol	21887	0,000	0,013	0,042	0,000	1,000	1,000	68
futbol	5720	0,000	0,035	0,065	0,000	2,000	1,600	69
futbol	947	0,000	0,157	0,157	0,000	1,348	1,000	70
futbol	10119	0,500	0,135	0,123	1,000	2,167	1,083	71
futbol	1875	0,000	0,075	0,073	0,000	1,250	1,000	72
futbol	1566	0,500	0,358	0,200	1,000	3,526	1,000	73
path	1356	0,000	0,098	0,087	0,000	1,437	1,096	74
path	2880	0,500	0,070	0,056	1,000	1,300	1,000	75
path	1323b	0,000	0,056	0,099	0,000	1,714	1,333	76
path	2956	0,500	0,167	0,073	1,000	2,067	1,000	77
path	3190	0,000	0,175	0,170	0,000	2,389	1,333	78
path	3663	0,500	0,113	0,103	1,000	1,000	1,000	79
path	3246	0,000	0,114	0,139	0,000	1,857	1,500	80
ghost	849	0,000	0,067	0,068	0,000	1,879	1,091	81
ghost	2322	0,000	0,307	0,110	0,000	5,200	1,000	82
ghost	1939b	0,500	0,074	0,116	1,000	1,400	1,375	83
ghost	1801	0,500	0,079	0,126	1,000	1,500	1,714	84
scary	1586b	0,500	0,066	0,035	1,000	2,800	1,000	85
ghost	1395	0,000	0,093	0,125	0,000	1,909	1,286	86
scary	908	0,000	0,064	0,071	0,000	1,644	1,090	87
scary	1025b	0,500	0,091	0,154	1,000	1,833	1,333	88
ghost	3829	0,000	0,144	0,100	0,000	3,444	1,000	89
scary	2248	0,500	0,079	0,111	1,000	1,200	1,200	90
scary	1380	0,500	0,078	0,081	1,000	2,500	1,000	91
scary	2316	0,000	0,129	0,105	0,000	2,455	1,500	92
scary	2464	0,000	0,220	0,310	0,000	2,310	1,160	93
futbol	16566	0,000	0,054	0,050	0,000	1,220	0,950	94
futbol	50677	0,000	0,121	0,096	0,000	1,313	1,000	95
futbol	21827	0,000	0,345	0,194	0,000	5,060	1,000	96
futbol	9878	0,500	0,066	0,050	1,000	1,300	1,000	97
futbol	8366	0,000	0,265	0,268	0,000	2,042	1,071	98
futbol	44726	0,500	0,309	0,385	1,000	2,193	1,035	99

---

<b>futbol</b>	9628	0,500	0,170	0,148	1,000	1,318	1,000	100
<b>futbol</b>	21262	1,000	0,350	0,432	1,000	2,056	1,000	101
<b>futbol</b>	26431	0,500	0,005	0,035	1,000	1,000	1,000	102
<b>scary</b>	1776	0,500	0,060	0,055	1,000	1,833	1,000	103
<b>scary</b>	3649	0,500	0,101	0,152	1,000	1,429	1,667	104
<b>futbol</b>	2386	0,500	0,037	0,072	1,000	1,143	1,000	105

## Evaluaciones

Evaluaciones												
Orden	GL	SD	CT	N	A	MB	PRB	JM	EM	GB	Prom	Desv
0	5	5	5	5	5	5	5	5	5	5	5	0,00
1	2	3	3	3	4	3	3	4	2	2	3	0,74
2	2	2	2	2	3	1	1	2	1	1	2	0,67
3	2	2	3	2	2	1	1	3	1	1	2	0,79
4	2	2	2	2	2	2	2	3	1	1	2	0,57
5	2	2	3	3	3	3	2	4	2	2	3	0,70
6	2	1	2	2	2	1	1	3	1	1	2	0,70
7	2	2	2	2	2	2	1	4	1	1	2	0,88
8	2	2	2	2	2	2	1	3	2	1	2	0,57
9	4	5	4	4	3	3	5	5	4	2	4	0,99
10	4	4	4	4	4	3	4	4	4	2	4	0,67
11	4	4	4	5	5	5	5	5	4	2	4	0,95
12	4	3	3	3	3	4	3	4	3	2	3	0,63
13	5	5	4	4	4	4	3	5	3	2	4	0,99
14	3	3	2	3	3	2	2	3	3	2	3	0,52
15	3	3	2	2	2	2	1	2	2	1	2	0,67
16	2	2	2	2	3	2	2	1	1	2	2	0,57
17	3	3	2	3	3	2	2	2	1	1	2	0,79
18	4	4	3	2	3	3	4	3	2	2	3	0,82
19	1	3	1	1	2	1	1	1	1	1	1	0,67
20	4	3	2	2	3	2	4	3	3	3	3	0,74
21	5	5	5	5	5	5	5	5	5	5	5	0,00
22	3	4	3	3	3	3	3	4	3	3	3	0,42
23	5	5	3	5	4	3	5	5	4	5	4	0,84
24	5	4	4	4	4	4	5	4	4	5	4	0,48
25	5	4	4	4	5	4	4	4	5	5	4	0,52
26	5	5	4	4	3	3	4	4	5	5	4	0,79
27	3	4	3	3	2	1	3	3	3	3	3	0,79
28	3	4	2	2	2	1	2	3	2	2	2	0,82
29	3	5	3	2	2	3	3	4	2	3	3	0,94
30	3	4	2	2	3	3	2	3	2	2	3	0,70
31	2	2	2	1	2	2	2	2	1	1	2	0,48
32	3	3	2	3	3	3	3	4	3	2	3	0,57
33	2	3	2	2	3	3	2	4	2	2	3	0,71
34	2	2	2	1	3	2	2	3	1	1	2	0,74
35	3	3	2	3	4	3	3	3	3	2	3	0,57
36	2	3	2	2	3	3	3	3	3	1	3	0,71
37	1	1	1	2	2	1	5	2	4	1	2	1,41
38	2	2	2	1	3	3	3	3	3	2	2	0,70
39	3	2	1	2	3	4	3	4	3	2	3	0,95
40	1	3	2	2	3	4	3	1	3	2	2	0,97
41	1	2	4	2	2	3	2	1	1	1	2	0,99
42	5	5	5	5	5	5	5	5	5	5	5	0,00
43	4	4	4	4	5	4	5	5	4	4	4	0,48
44	5	5	3	4	4	4	5	5	5	5	5	0,71
45	3	4	3	4	4	4	3	4	3	3	4	0,53
46	4	4	3	3	5	4	3	3	3	3	4	0,71
47	3	4	2	3	4	3	3	4	3	3	3	0,63

48	2	3	3	1	2	2	1	3	1	1	2	0,88
49	3	4	4	3	3	2	3	3	2	2	3	0,74
50	3	4	3	2	3	2	4	3	2	2	3	0,79
51	3	4	3	3	4	4	2	4	2	2	3	0,88
52	4	4	3	3	4	4	3	4	4	3	4	0,52
53	2	3	2	4	3	3	2	3	1	2	3	0,85
54	1	3	3	2	3	2	2	2	2	2	2	0,63
55	2	3	4	2	3	2	4	3	1	2	3	0,97
56	3	3	2	2	4	3	4	4	5	3	3	0,95
57	2	3	3	2	2	2	3	3	3	3	3	0,52
58	2	3	2	3	3	3	3	4	2	4	3	0,74
59	1	1	1	1	1	1	1	1	1	1	1	0,00
60	2	4	3	3	4	4	3	5	4	4	4	0,84
61	3	4	2	3	3	3	3	4	4	4	3	0,67
62	2	3	3	2	2	2	2	3	2	2	2	0,48
63	5	5	5	5	5	5	5	5	5	5	5	0,00
64	4	5	4	5	4	5	5	5	5	5	5	0,48
65	2	2	2	2	3	2	2	1	1	1	2	0,63
66	3	4	2	3	3	3	4	4	4	3	3	0,67
67	2	3	1	1	3	2	2	2	2	2	2	0,67
68	2	3	3	2	3	3	3	1	4	3	3	0,82
69	4	5	4	4	4	5	5	5	5	5	5	0,52
70	2	2	3	2	3	3	3	4	1	2	3	0,85
71	3	3	3	2	3	3	3	4	4	3	3	0,57
72	1	2	2	1	2	2	2	1	1	2	2	0,52
73	1	2	2	1	2	2	4	1	3	2	2	0,94
74	4	4	4	4	5	4	5	3	5	3	4	0,74
75	3	3	2	2	4	4	2	2	2	1	3	0,97
76	4	3	4	3	3	5	4	4	5	4	4	0,74
77	1	1	1	1	2	2	1	1	1	1	1	0,42
78	3	1	1	1	2	2	2	1	1	1	2	0,71
79	3	3	1	2	3	2	2	1	2	2	2	0,74
80	3	3	2	1	3	2	2	3	2	2	2	0,67
81	3	3	3	3	3	3	3	3	3	3	3	0,00
82	2	2	2	1	3	2	3	3	3	3	2	0,70
83	4	3	3	2	2	2	2	4	3	2	3	0,82
84	4	4	3	2	2	1	3	5	2	4	3	1,25
85	4	4	3	2	2	3	4	5	4	4	4	0,97
86	2	3	3	3	4	3	3	4	4	3	3	0,63
87	3	3	3	2	3	3	4	4	4	3	3	0,63
88	2	2	2	2	3	3	3	4	3	2	3	0,70
89	2	3	2	1	3	3	3	3	2	3	3	0,71
90	2	3	2	2	3	3	2	4	1	2	2	0,84
91	3	3	3	2	3	3	2	3	2	1	3	0,71
92	1	2	2	1	3	2	2	2	1	1	2	0,67
93	1	2	1	1	3	1	3	1	1	2	2	0,84
94	3	5	2	2	3	3	3	4	2	2	3	0,99
95	2	4	2	2	3	2	3	3	2	2	3	0,71
96	1	1	1	1	1	1	1	1	2	1	1	0,32
97	1	2	1	1	3	3	2	3	1	1	2	0,92
98	1	1	2	1	1	1	3	1	2	2	2	0,71
99	1	2	2	1	3	2	2	1	3	2	2	0,74

---

100	1	2	2	2	2	2	2	1	2	3	2	0,57
101	1	2	2	2	3	1	2	1	3	3	2	0,82
102	2	2	2	3	2	2	3	2	2	2	2	0,42
103	2	2	2	2	2	2	2	2	2	3	2	0,32
104	2	2	1	2	3	2	2	2	1	1	2	0,63
105	2	2	2	2	3	3	3	3	1	2	2	0,67

**ISSN 1510-7264**

Nicolás de León

Maestría en Ingeniería en Computación

Facultad de Ingeniería

Universidad de la República

Montevideo, Uruguay, 2010